Downloaded from orbit.dtu.dk on: Jan 22, 2025

DTU Library

Perception for Precision Agriculture Robotics using Deep Learning

Güldenring, Ronja

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Güldenring, R. (2024). *Perception for Precision Agriculture Robotics using Deep Learning*. Technical University of Denmark.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Perception for Precision Agriculture Robotics using Deep Learning

by
Ronja Güldenring

Dissertation submitted February 2024

Ph.D. Thesis
Doctor of Philosophy

DTU Electro
Department of Electrical and Photonics Engineering

Perception for Precision Agriculture Robotics using Deep Learning

Ronja Güldenring

Kongens Lyngby 2024

DTU Electro
Department of Electrical and Photonics Engineering
Automation & Control
Technical University of Denmark

Ørsteds Plads
Bygning 343
2800 Kongens Lyngby, Denmark
https://electro.dtu.dk/

English Summary

The world population is expected to increase drastically in the next 30 years. Accordingly, we need to increase food production while treating the limited available land sustainably and keeping the environmental footprint low to decelerate global warming. Organic farming on a large scale is challenging because of the cost-intensive high demand of manual and physically demanding labor.

Digitalization can pave the way towards more sustainable agriculture. Hereby, agricultural precision robots constitute one solution to enable large-scale organic farming with high yield while having a substantially lower environmental influence. The robots are equipped with sensors and actuators to perceive the fields and perform selective plant treatments. The shortage of workers for physically demanding field labor can be mitigated by robots that have the potential to be scaled up by running multiple robots both day and night.

Therefore, agricultural robots in crop fields have gained great attention in recent years. Weed control is performed for plants of small growth stages, appearing predominantly isolated on soil background. On the contrary, grassland fields of dairy farming are less researched and introduce further challenges. There, weed plants appear on a cluttered background, consisting of other vegetation such as grass or clover. Weed identification becomes more challenging because we have to deal with more occlusions. Furthermore, foreground and background vegetation share similar pixel colors and chlorophyll content.

In this work, we develop the perception system for a field robot operating in grassland fields. We target the most problematic grassland weed *Rumex obtusifolius L.*, which spreads heavily when not treated. It leads to bad field yield and can be toxic for livestock when large amounts are consumed. The field robot applies experimental herbicide-free weeding using laser and electrocution techniques.

We exclusively use deep learning approaches, namely convolutional neural networks, to tackle the visual identification and analysis of *Rumex* weeds. Robust deep learning models come at the cost of large amounts of data, whereas ground-truth annotations are expensive to obtain. We investigate approaches to reduce this cost and improve label efficiency. We generate plant-specific image compositions and use the synthetic images during supervised model training. Moreover, we obtain domain-

specific representations with unlabeled data using self-supervised contrastive learning. Finally, we make two grassland datasets publicly available for the community: (1) *RumexWeeds* containing broad view images with coarse weeds annotations and (2) *RumexLeaves* containing close-up images of the weeds and fine-grained annotations of plant traits. Based on those datasets, we develop different perception models enabling our modular field robot to remove *Rumex* weeds efficiently. First, the weed and its approximate joint stem are detected on a broad-view image. The robot navigates toward the detected weed to position the weeding unit above the plant. A close-view image of the weed is taken, allowing for the extraction of more detailed plant traits, such as leaf, stem, and vein instances, enabling a more efficient weeding procedure but also allowing for automated detailed phenotyping.

This is a cumulative thesis and it is based on six papers. The first five papers are peer-reviewed and published in different robotics and agriculture venues/journals. The approach for generating plant-specific image compositions was a finalist for the best paper award on agri-robotics at the *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. The last paper is an integration paper and presents the final outcome of the joint work achieved within the underlying European project GALIRUMI. It is still under preparation and will be submitted soon.

Danish Summary

Verdens befolkning forventes at stige drastisk i løbet af de næste 30 år. Derfor er vi nødt til at øge fødevareproduktionen, samtidig med at vi behandler den begrænsede tilgængelige jord bæredygtigt og holder det miljømæssige fodaftryk lavt for at bremse den globale opvarmning. Økologisk landbrug i stor skala er en udfordring på grund af det omkostningskrævende, hårde manuelle arbejde.

Digitalisering kan bane vejen for et mere bæredygtigt landbrug. Præcisionsrobotter til landbruget er en potentiel løsning for økologisk landbrug i stor skala med højt udbytte og samtidig en væsentligt lavere miljøpåvirkning. Robotterne er udstyret med sensorer og aktuatorer til at opfatte markerne og udføre selektive plantebehandlinger. Manglen på arbejdskraft til det fysisk krævende markarbejde kan ligeledes afhjælpes med robotter, der nemt kan skaleres op ved at køre med flere robotter både dag og nat.

Landbrugsrobotter i afgrødemarker har fået stor opmærksomhed i de senere år. Ukrudtsbekæmpelse udføres for planter i små vækststadier, der overvejende forekommer isoleret på jordbaggrund. I modsætning hertil er græsarealer inden for mælkeproduktion mindre undersøgt og udgør yderligere udfordringer. Ukrudtsplanter optræder på en baggrund, der består af anden vegetation som græs eller kløver. Ukrudtsidentifikation bliver mere udfordrende grundet flere okklusioner. Desuden deler forgrunds- og baggrundsvegetation lignende pixelfarver og klorofylindhold.

I dette projekt udvikler vi en perceptionssystem til en markrobot, der arbejder på græsmarker. Vi fokuserer på det mest problematiske græsukrudt *Rumex obtusifolius L.*, som spreder sig voldsomt, når det ikke behandles. Det fører til dårligt markudbytte og kan være giftigt for dyr, når store mængder indtages. Markrobotten fjerner udkrudt ved hjælp af eksperimentelle laser- og elektriske teknikker.

Vi bruger udelukkende deep learning, navnlig convolutional neural networks, til at realisere den visuelle identifikation og analyse af *Rumex* ukrudt. Robuste deep learning-modeller behøver en stor mængde data, mens ground-truth-annotationer er dyre. I forbindelse med oprettelsen af vores datasæt undersøger vi muligheder til at forbedre label-effektiviteten. Vi laver plantespecifikke billedsammensætninger og bruger de syntetiske billeder under superviseret modeltræning. Desuden opnår vi domænespecifikke repræsentationer med umærkede data ved hjælp af self-supervised contrastive learning. Endelig gør vi to datasæt om græsarealer offentligt tilgængelige

for samfundet: (1) RumexWeeds, der indeholder billeder i bredformat med grove ukrudtsannotationer, og (2) RumexLeaves, der indeholder nærbilleder af ukrudtet og detaljeret annotationer af planteegenskaber. Baseret på disse datasæt udvikler vi forskellige opfattelsesmodeller, der gør det muligt for vores modulære markrobot at fjerne *Rumex* ukrudt effektivt. Først registreres ukrudtet og dets plantecenter på et bredt billede. Robotten navigerer hen mod det registrerede ukrudt for at placere lugeenheden over planten. Der tages et nærbillede af ukrudtet, som gør det muligt at udtrække mere detaljerede planteegenskaber som f.eks. blade, stængler og årer, hvilket muliggør en mere effektiv lugeprocedure, men også automatiseret detaljeret fænotypebestemmelse.

Dette er en kumulativ PhD thesis, og den er baseret på seks artikler. De første fem artikler er peer-reviewed og udgivet i forskellige konferencer eller journals inden for robotteknologi og landbrug. Metoden til at generere plantespecifikke billedkompositioner var finalist til prisen for bedste artikel om landbrugsrobotik på konferencen *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Den sidste artikel er et integrationsartikel og præsenterer det endelige resultat af det fælles arbejde, der er opnået i det underliggende europæiske projekt GALIRUMI. Den er stadig under forberedelse og vil snart blive indsendt til en journal.

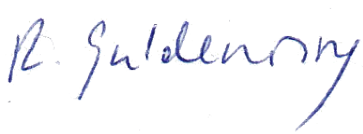
Preface

This Ph.D. thesis was prepared at the Department of Electrical and Photonics Engineering in the Automation and Control Group at the Technical University of Denmark to fulfill the requirements for acquiring the Ph.D. degree. It was supervised by Professor Lazaros Nalpantidis and co-supervised by Associate Professor Evangelos Boukas and Professor Emeritus Ole Ravn. From April 2023 to September 2023 an external research stay was carried out at the Department of Plant and Environmental Sciences at the University of Copenhagen under Jesper Svensgaard.

The work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258.

During the course of the Ph.D. project, eight peer-reviewed papers have been published. One additional article is under preparation and will be submitted soon, thus, in this thesis a preliminary preprint is provided. Six of the collection of papers are relevant to the Ph.D. project and form the basis for the cumulative thesis. They are provided to the reader at the end of the document as paper A to paper F.

Kongens Lyngby, February 13, 2024



Ronja Güldenring

Acknowledgements

At first, I would like to thank my main supervisor, Lazaros Nalpantidis, and my co-supervisors, Evangelos Boukas and Ole Ravn, for your guidance and support throughout the project. Lazaros, I especially appreciate the immense trust and care you put into your Ph.D. students. Clearly, you act in the interest of the students.

Thanks to Jesper Svensgaard and the tech team for a warm welcome at the Department of Plant and Environmental Sciences at the University of Copenhagen.

Thanks to my close co-authors and GALIRUMI project partners for the great collaborations: Lazaros Nalpantidis, Evangelos Boukas, Ole Ravn, Frits K. van Evert, Rasmus Eckholdt Andersen, Patrick Schmidt, Jiahao Li, Ard Nieuwenhuizen, Jacobo Domínguez, and Esther López.

I am immensely grateful to the people I know longest in life: my parents and siblings. I know, that I can always count on your support, whatever challenge I might face. Thanks to my parents for supporting me during my educational path financially and advisory. It took me some time to find out what I wanted to do professionally: I started with Bachelor's studies in Environmental Engineering, ended up with a Bachelor's degree in Mechanical Engineering, and continued my Master's studies in Computer Science. My parents have always stood by my side and helped me make good decisions by talking the options through. In my opinion, my PhD project covers all my values and interests at once, where I had the pleasure to contribute to society in a meaningful way with highly interesting technology.

Thanks to my beloved child for taking care of my work-life balance. She made sure that I do not work overtime, but rather enjoy afternoons at the playground.

Thanks to my partner and best friend Eike for your love and your efforts in making me a confident person. You were always up-to-date with my work and the associated challenges. I highly appreciate our inspiring and effective brainstorming sessions throughout the years. However, a joint paper remains a couple's goal for the future.

Publications

All conference papers and journal articles are peer-reviewed and published except for item F, which is still under preparation and will be submitted soon.

Included in the thesis

Conference Papers

A **Ronja Güldenring**, Evangelos Boukas, Ole Ravn, and Lazaros Nalpantidis, *Few-leaf learning: Weed segmentation in grasslands.* Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. **Finalist for the Best Paper Award on Agri-Robotics**.

Journal Articles

B **Ronja Güldenring**, Lazaros Nalpantidis, *Self-supervised contrastive learning on agricultural images.* 191, 106510, Computers and Electronics in Agriculture, 2021

C **Ronja Güldenring**, Frits K. van Evert and Lazaros Nalpantidis, *RumexWeeds: A grassland dataset for agricultural robots.* Journal of Field Robotics, 2023

D Jiahao Li, **Ronja Güldenring**, and Lazaros Nalpantidis, *Real-time joint-stem prediction for agricultural robots in grasslands using multi-task learning.* vol. 13, no. 9, Agronomy, 2023

E **Ronja Güldenring**, Rasmus Eckholdt Andersen, Lazaros Nalpantidis, *Zoom in on the plant: Fine-grained analysis of leaf, stem and vein instances.* 1588–1595, IEEE Robotics and Automation Letters, 2024

F **Ronja Güldenring**, Jacobo Domínguez, Ayodeji Deolu-Ajayi, Menno Sytsma, Esther López, Julien Fradin, Hans Pekkeriet, Harry Koonstra, Annelies Beniers, Anouk Leunissen,Ard Nieuwenhuizen, Lazaros Nalpantidis, Frits K. van Evert, *Smart implements for robotic precision agriculture: A case study on laser and electric weeding in grasslands*, 2024, **under preparation**

Datasets

G **Ronja Güldenring**, Frits K. van Evert and Lazaros Nalpantidis, *Data for RumexWeeds: A grassland dataset for agricultural robotics*. URL: https://data.dtu.dk/articles/dataset/Data_for_RumexWeeds_A_Grassland_Dataset_for_Agricultural_Robotics_/17040518, 2023

H **Ronja Güldenring** and Lazaros Nalpantidis, *RumexLeaves*. URL: https://data.dtu.dk/articles/dataset/_strong_RumexLeaves_Dataset_introduced_by_Paper_Fine-grained_Leaf_Analysis_for_Efficient_Weeding_Robots_strong_/23659524, 2023

Not included in the thesis

Conference Papers

I Michael Rudolph, Youssef Dawoud, **Ronja Güldenring**, Lazaros Nalpantidis, Vasileios Belagiannis, *Lightweight Monocular Depth Estimation through Guided Decoding*. Proceedings of International Conference on Robotics and Automation (ICRA), 2022

J Patrick Schmidt, **Ronja Güldenring**, Lazaros Nalpantidis, *SIFT-guided Saliency-based Augmentation for Weed Detection in Grassland Images: Fusing Classic Computer Vision with Deep Learning*. Proceedings of International Conference on Computer Vision Systems (ICVS), 2023

K **Ronja Güldenring**, Michael Görner, Norman Hendrich, Niels Juul Jacobsen, Jianwei Zhang, *Learning local planners for human-aware navigation in indoor environments*. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020

Challenge Participation

L **Ronja Güldenring**, Lazaros Nalpantidis, **3rd place award** with *Mask2Former@PhenoBench*. Hierarchical Panoptic Segmentation of Crops and Weeds, 8th workshop on Computer Vision in Plant Phenotyping and Agriculture, International Conference on Computer Vision (ICCV), 2023

Contents

English Summary i

Danish Summary iii

Preface v

Acknowledgements vii

Publications ix
Included in the thesis . . . ix
Not included in the thesis . . . x

Contents xi

List of Abbreviations xiii

1 Introduction 1
1.1 Scope . . . 2
1.2 Goal and main contributions . . . 4
1.3 Thesis structure . . . 5

2 Background 7
2.1 Foundations of deep learning . . . 7
2.1.1 Layer types . . . 8
2.1.2 Loss functions . . . 9
2.1.3 Optimizers . . . 10
2.1.4 Regularization methods . . . 12
2.2 Applications of deep learning in visual perception . . . 13
2.2.1 Metrics . . . 13
2.2.2 Backbone networks . . . 15
2.2.3 Object detection approaches . . . 17
2.2.4 Image segmentation approaches . . . 19
2.2.5 Keypoint detection approaches . . . 19
2.2.6 Contrastive self-supervised learning . . . 20

3 Related Work **23**
3.1 Agricultural weed datasets . . . 23
3.1.1 Grassland field datasets . . . 23
3.1.2 Crop field datasets . . . 24
3.1.3 Agricultural robotics datasets . . . 26
3.2 Representation learning in the domain of agriculture . . . 27
3.3 Weed detection . . . 28
3.4 Fine-grained plant analysis . . . 29
3.4.1 Estimation of leaf count, -size and -shape . . . 30
3.4.2 Joint stem detection . . . 30
3.4.3 Vein detection . . . 31

4 Contributions **33**
4.1 The robot platform . . . 33
4.2 Rumex dataset creation . . . 35
4.3 Rumex weed detection . . . 40
4.4 Fine-grained rumex weed analysis . . . 41

5 Conclusion & Future Outlook **45**
5.1 Conclusion . . . 46
5.2 Future Outlook . . . 46

Bibliography **49**

A Few-leaf learning: Weed segmentation in grasslands **63**

B Self-supervised contrastive learning on agricultural images **81**

C RumexWeeds: A grassland dataset for agricultural robots **111**

D Real-time joint-stem prediction for agricultural robots in grasslands using multi-task learning **141**

E Zoom in on the plant: Fine-grained analysis of leaf, stem and vein instances **159**

F Smart implements for robotic precision agriculture: A case study on laser and electric weeding in grasslands **179**

List of Abbreviations

AP Average Precision

CNN Convolution Neural Network

CSP Cross Stage Partial

CSRT Discriminative correlation filter with channel and Spatial Reliability Tracker

CVAT Computer Vision Annotation Tool

DC Deformable Convolutions

DTU Technical University of Denmark

FN False Negatives

FP False Positives

GALIRUMI GALIleo-assisted robot to tackle the weed RUMex obtusifolius and increase the profitability and sustainability of dairy farming

GNSS Global Navigation Satellite System

GPS Global Positioning System

IMU Inertial Measurement Unit

IoU Intersection over Union

IROS IEEE/RSJ international conference on intelligent RObots and Systems

JI Jaccard Index

mAP mean Average Precision

NCE Noise Contrastive Estimation

SDG Stochastic Gradient Descent

SSL Self-supervised Learning

OBB Oriented Bounding Box

OKS Object Keypoint Similarity

POKS Projected Object Keypoint Similarity

RGB Red Green Blue

TP True Positives

TN True Negatives

CHAPTER 1

Introduction

The increasing world population is one of the major challenges our society is facing. According to Dijk et al. [Dij+21], food demand is expected to increase by 35 - 65% in the period 2010 - 2050, whereas hunger is likely to increase by 8%.

The majority of agricultural land is not farmed organically. Agrochemicals such as herbicides, pesticides, and fertilizers are used to control weeds, prevent plant diseases, and perform artificial fertilization. These actions lead to increased field yield, in the short run. Moreover, monocultures have become established because it is cheap to specialize in one crop, in the short run.

The usage of agrochemicals and monocultures results in land degradation, groundwater pollution, and reduced biodiversity which heavily impacts the environmental footprint. In the long run, those side effects will lead to expensive food production and degraded field yield.

Organic farming aims to perform farming with a low environmental footprint. It often comes at the cost of higher expenses and lower field yield [Smi+19]. Hereby, weed control is one of the biggest challenges; manual weeding is often necessary, but it is expensive, physically demanding, and laborers are often hard to find. As a result, it becomes challenging for conventional farms to shift towards organic practices.

In food production, the available land is a limited resource and has not been treated adequately in recent decades. We need to increase field yield while preserving and even supporting the underlying ecosystem to ensure long-term healthy food production. Precision agricultural robots constitute one potential solution for combining sustainable farming with high yield. Robots are envisioned as field workers that can work autonomously and continuously day and night. They can be equipped with a variety of extension modules consisting of relevant sensors and actuators to perform different tasks, such as monitoring, seeding, weed control, disease control, and harvesting. Selective plant treatment will play a major role, as the perception granularity and the actuator capabilities improve. Recent progress will allow for mixed cropping [Cou+22; Gar+22] containing diverse plant compositions and distributions, where different plant species are handled individually depending on their visual appearance and neighborhood distribution. Another example of selective plant treatment is to keep a certain percentage of weed plants, which can contribute positively to insect biodiversity and soil quality, while not substantially suppressing the crop

yield [KK23].

1.1 Scope

In the scope of this thesis, we target the agricultural domain of grassland fields for dairy farming. These fields host grazing animals, but they also get regularly mowed, whereas the field yield serves as food for livestock.

Within dairy farming, *Rumex obtusifolius L.* (broad-leaved dock) and *Rumex crispus L.* (curled dock) are among the most troublesome grassland weeds worldwide. They can grow in a wide range of climatic conditions as well as in many different soil types. As an example, approximately 60 - 70% of grassland fields both in Germany and Japan are heavily infested with both Rumex species [Zal04; Hon86]. Furthermore, it is stated that in Europe, 80% of the herbicides used for weed control in grassland are applied to *Rumex obtusifolius L.* [Zal04]. *Rumex* is less nutritious than grass for grazing animals, and if consumed in large doses it can get poisonous for the animals, due to the large proportion of nitrate and oxalate in the plant [S H04]. *Rumex* shares soil resources and therefore, competes with more nutritious and sown pasture species [Zal04]. If *Rumex* is not controlled, it creates a significant yield loss of 10% - 40% [Eve+11]. The taproot can grow huge: up to 40 mm thick and up to 2.5m into soil [Zal04], which makes it challenging to remove *Rumex* properly with non-chemical tools.

(a) **(b)**

Figure 1.1: This is *Rumex obtusifolius L.*. (a) An organic grassland field is infested with *Rumex* weeds. The weeds have already grown largely and are flowering with their brown/red seed stems popping out. (b) Example top-down images of *Rumex* appearing in different conditions such as growth stage, lightning conditions, and background.

The European project GALIRUMI supporting this work explores two different ways of controlling *Rumex obtusifolius/crispus L.* sustainably and autonomously: (1) laser weeding and (2) electric weeding. Laser weeding is applied to remove the leaves of the plant above the soil. It must be performed multiple times throughout the year because only frequent defoliation can effectively control *Rumex* weeds [Eve+20]. On the contrary, electric weeding applies a high-voltage current to the plant. The expectations are, that not only the plant is killed above ground, but fully destroyed deep down to the tap root. This technique is expected to be more effective and must only be applied once a year. Weeding implements for each control techniques can be attached to the same commercially available agriculture ground vehicle, as shown in Figure 1.2. A summary of the outcome of the GALIRUMI project can be found in paper F. We also refer the reader to the demonstration video of the GALIRUMI robot for a visual overview `https://youtu.be/3LOMdMfIjOs` .

Figure 1.2: The GALIRUMI robot is the commercial CEOL robot platform from Agreenculture, extended with two different implements for laser and electric weeding. Both are equipped with the same visual perception system operating in two stages, whereas in stage one, weeds and their approximate joint stems are identified in a broad-view image in real-time, and in stage two, fine-grained plant analysis is performed on a close-up view to enable device-specific weed intervention.

Our developed visual perception system is generic and can be paired with both weeding implements. It operates in two stages, whereas in the first stage, the weed is identified in the field on a coarse level. Hereby, we set a focus on the data throughput, because we want to process the field at a certain speed with a high inference rate to allow the robot to cover a reasonable amount of hectares per hour. Once a weed is detected, the robot navigates toward the weed to position the weeding unit above. A second camera provides a close-up view of the weed to allow a fine-grained analysis of the plant. The branching system of the plant as well as the individual leaves are

identified, enabling precise and device-specific interventions of the weeding unit with the plant.

1.2 Goal and main contributions

The *goal* of this Ph.D. project is to build a visual perception system that extracts plant information on different granularity levels, from coarse detection information to fine-grained plant traits. Exemplarily, it is developed for the analysis of the problematic grassland weed *Rumex obtusifolius L.* and aims to enable GALIRUMI's advanced organic weeding methodologies, nevertheless, it has usability in other domains such as on-field phenotyping and high-resolution plant monitoring for mixed crop fields.

A short summary of our papers is provided in the following. For all papers, the most recent version has been attached to the main document in paper A to paper F, reformatted in order to align the style of the main document.

Paper A was a finalist for the best paper award on agri-robotics at the International Conference on Intelligent Robots and Systems (IROS 2021). We build on the observation that plant characteristics of images collected within one day and field do not vary substantially. We use a very small number of manually annotated leaves to compose new plants and paste the composed plants on new diverse background images. A segmentation model is trained on the synthetically generated images. We experimentally show, that the majority of plants in real images can be segmented with a certain accuracy.

In paper B, we explore a state-of-the-art contrastive self-supervised approach SwAV [Car+20] for the first time in the domain of agriculture. We use SwAV to obtain domain-specific pre-trained weights and compare their effectiveness to ImageNet pre-trained weights by transferring them to different classification and segmentation downstream tasks. Our results show, that pre-training with SwAV is particularly effective for classification tasks with a limited amount of annotated images.

In paper C, we make a dataset, *RumexWeeds*, publicly available, which targets the detection of the grassland weeds: *Rumex obtusifolius L.* and *Rumex crispus L.*. It is one of the first larger agricultural robotics datasets including annotated image data in grasslands. *RumexWeeds* includes whole image sequences with totally 5,510 images of 2.3 MP resolution and 15,519 manual bounding box annotations as well as 340 ground truth pixels-wise annotations, collected at 3 different farms and 4 different days in summer and autumn 2021. Additionally, navigational robot sensor points from GNSS, IMU, and odometry are recorded. The paper also provides baseline detection results with YOLOX [Ge+21] and UNet [RPB15], which are expected to be improved by the research community.

In paper D, we predict the joint stem position along with the enclosing bounding box for *Rumex* in grasslands. Two YOLO-based approaches are compared to each other, which have architectural differences in the number of stages and the way of predicting keypoints with either classification via heatmaps or regression. We show

that the integration of the additional task of joint stem detection has a positive and boosting effect on the weed detection performance. The dataset *RumexWeeds* is extended with joint stem annotations which are provided publicly.

In paper E, a fine-grained plant analysis in real-world fields is performed by predicting leaf instances and corresponding polylines for the stem and primary vein of each leaf in single RGB images. We introduce an adapted evaluation metric that complies with keypoint-guided polylines, consisting of distinct keypoints and pseudo keypoints, which may lie at any point on neighboring line segments. The detailed plant analysis enables advanced automated phenotyping as well as power-efficient weeding strategies. Again, the accompanying dataset with the expensive annotations *RumexLeaves* is made public.

Paper F is an integration paper and presents the final outcome of the joint work achieved in the underlying European project GALIRUMI. In this thesis, it serves as a detailed summary of the agricultural application behind this thesis. The work is still under preparation and has not yet been submitted for publication.

Note, that in chapter 4, a more detailed overview of our contributions is given, providing the big picture of the work and outlining the interconnections between the papers.

1.3 Thesis structure

The thesis structure is as follows.

In chapter 2, fundamental concepts of deep learning are revisited including relevant layer types, loss functions, optimizers, and regularization methods. Moreover, we explain relevant deep learning architectures in the domain of visual perception. Hereby, we focus on architectures that have been considered and modified in our papers.

In chapter 3, we summarize relevant literature at the intersection of deep learning, computer vision, and weed management in real-world fields. The chapter merges references from all papers in a meaningful ensemble, enriched with new articles, which have followed at a later point in time.

In chapter 4, a detailed overview of our papers is given, embedded in the big picture of the project and outlining the interconnections between the papers.

Chapter 5 concludes the thesis. The Ph.D. project is summarized, followed by future perspectives for the work.

Finally, the six articles that form the basis for this thesis are appended at the end of the document as paper A to paper F.

CHAPTER 2

Background

In this chapter, we revisit fundamental concepts, that form the foundation for our papers in chapter 4. In section 2.1, we summarize the foundations of deep learning including the general learning process as well as relevant layer types, loss functions, optimizers, and regularization techniques. In section 2.2, we set a stronger focus on the application of deep learning in the visual domain. At first, we present relevant metrics to evaluate vision models. Then, we present established network architectures for different tasks such as classification, object detection, image segmentation, keypoint detection, and self-supervised learning. Most of the presented network architectures are further used within our papers.

2.1 Foundations of deep learning

When training a neural network Θ, the goal is to find the best parameter set Θ_B to approximate the objective function. The learning process is iterative and consists of four main steps described in the following.

Forward pass. The input X is forwarded through the network architecture. The architecture consists of one or multiple layer types, while the most relevant are introduced in subsection 2.1.1. The output of the network is $\hat{Y} = f(X, \Theta)$.

Loss. The prediction $\hat{Y}$ and the true output Y are compared by computing the loss $\mathcal{L}(\Theta, \hat{Y}, Y)$. The choice of loss function depends on the learning objective. In subsection 2.1.2, we present the relevant loss functions for this thesis.

Backward pass. The global gradient of loss $\nabla_\Theta L(\Theta)$ is back-propagated through the neural network, flowing in the opposite direction of the forward pass. The chain rule is applied to compute a local gradient for each neuron and update it locally.

Update. The weights of all layers are updated according to their local gradients. Different optimizers use different strategies on how to update the weights accordingly. Subsection 2.1.3 presents the ones relevant for this thesis.

2.1.1 Layer types

In the following, the most common and relevant layer types are presented.

Fully-connected layer. All neurons of the outgoing layer are connected to all neurons of the ingoing layer. The neurons are densely connected and its computation gets expensive for large input data. [GBC16]

Activation layer. Activation layer introduces non-linearity to neural networks, allowing for non-linear function approximation. There exist a variety of different activation functions, that can be applied to the input. The most common ones are the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).
The sigmoid function maps the input between $[0, 1]$: $sigm(x) = \frac{1}{1+e^{-x}}$.
The tanh function zero-centers the sigmoid function, resulting in output values between $[-1, 1]$.
The ReLU function maps negative input to 0, while positive input remains the same: $ReLU(x) = max(0, x)$. Compared to sigmoid and tanh, ReLU has a non-saturating form, preventing the problem of vanishing gradients and leading to faster gradient convergence. [GBC16]

Convolutional layer. Multiple square filters f_i with $i \in \{0, 1, ..., Ch\}$ and of size $[m \times m]$ with $m = 2k + 1$ are slid over the input g and for each position (x, y) the discrete convolutional operation is computed as shown in the following equation.

$$h_i[x, y] = f_i * g[x, y] = \sum_{u=-k}^{k} \sum_{u=-k}^{k} f[u, v]g[x - u, y - v] \tag{2.1}$$

The output of the individual filters is stacked, resulting in an activation map of size $W \times H \times Ch$. Convolutional layers with small filter sizes m can be computed efficiently and are therefore widely used for large input data such as images. [GBC16]

Deformable convolutional layer. The applied filter is not a fixed grid, but instead, the individual filter weights are applied with a learnable offset. The offset depends on the input feature map and is therefore recalculated for each input tensor. The additional offset learning increases the number of model parameters and hence, the computational costs. The advantage of deformable convolution (DC) is that it increases the receptive field and allows the model to learn features spatially distant from the center position (x, y). [Zhu+18]

Pooling layer. Discrete filters are applied to reduce the input dimension. Max- and average-pooling are the most common variants, where the maximum or average is taken from the receptive field of size $[m \times m]$. [GBC16]

Upsampling layer. The input is scaled up in width and height by repeating the

columns and rows accordingly. Hence, the output doesn't contain different information. It is usually followed by a standard convolutional layer.

Transposed convolutional layer. The transposed convolutional layer scales up the input width and height with learnable filters. Strictly seen, it is a convolutional layer. However, before applying the convolutional operation, the input is enlarged with padding to achieve the desired dimensionality increase.

Residual layer. Connections are introduced to skip certain layers, i.e. the identity input is directly added to the output of the skipped layers. Residual layers potentially prevent the problem of vanishing gradients and allow for the training of deep networks, consisting of a large number of hidden layers. [He+16a]

2.1.2 Loss functions

In this section, we will give an overview of all loss functions relevant to our papers A to E. All loss functions are presented for a mini-batch of B datapoints. We present regression loss functions, which handle continuous output values as well as classification loss functions, dealing with discrete values, such as C different classes.

L1 loss. The regression function $\mathcal{L}_1$ computes the mean absolute error between the prediction $\hat{Y}$ and the true output Y.

$$\mathcal{L}_1 = \frac{1}{B}\sum_{n=1}^{B} |\hat{Y} - Y| \tag{2.2}$$

MSE loss. The mean squared error is computed between the prediction $\hat{Y}$ and the true output Y. It is a regression loss.

$$\mathcal{L}_{MSE} = \frac{1}{B}\sum_{b=1}^{B} (\hat{Y} - Y)^2 \tag{2.3}$$

Cross-entropy loss. The classification loss function measures the difference between the true and predicted probability distributions ($p_c, \hat{p}_c$ with $c \in \{1, ..., C\}$,). For each datapoint belonging to class c ($Y_c = 1$), the logarithm is computed for the predicted probability belonging to the same class $\hat{p}_c$.

$$\mathcal{L}_{CE} = -\frac{1}{B}\frac{1}{C}\sum_{b=1}^{B}\sum_{c=1}^{C} Y_c \log \hat{p}_c \tag{2.4}$$

Focal loss. The focal loss extends the cross-entropy loss with an adaptive weighting factor $(1-\hat{p}_c)$. It down-weighs easy datapoints and increases the focus on the difficult ones. Note that γ and α_c are hyperparamters. [Lin+17]

$$\mathcal{L}_{FOCAL} = -\frac{1}{B}\frac{1}{C}\sum_{b=1}^{B}\sum_{c=1}^{C} Y_c \alpha_c (1-\hat{p}_c)^{\gamma} \log \hat{p}_c \tag{2.5}$$

Lovasz-Softmax loss. The Lovasz-Softmax loss function is particularly relevant for image segmentation tasks as it optimizes directly towards the Jaccard Index introduced in Equation 2.23. The Jaccard Index itself is not differentiable and can therefore not be used during optimization. Instead, the lovasz-hinge applied to the Jaccard Index $\overline{\nabla_{JI}}$ serves as surrogate. [YB15]

$$\mathcal{L}_{LS} = -\frac{1}{B}\frac{1}{C}\sum_{b=1}^{B}\sum_{c=1}^{C} \overline{\nabla_{JI}}(m(c)) \tag{2.6}$$

m(c) is a misclassification vector, which contains the probability of not belonging to class c.

$$m_i(c) = \begin{cases} 1-\hat{p}_c \text{ if } Y_c == 1 \\ \hat{p}_c \text{ otherwise} \end{cases} \tag{2.7}$$

InfoNCE loss. The Info Noise Contrastive Estimation is a loss function for self-supervised contrastive learning. The embedding of the query sample z_q is contrasted towards the embedding of a positive sample z_p and N negative samples $(z_1, ..., z_N)$. The logarithm is computed for the softmax of the vector similarity of the positive pair over the sum of vector similarities of all pairs. Note, that τ is a hyperparameter. [OLV18]

$$\mathcal{L}_{NCE} = -\frac{1}{B}\sum_{b=1}^{B} \log \frac{\exp\left(f_{sim}(z_q, z_p)/\tau\right)}{\exp\left(f_{sim}(z_q, z_p)/\tau\right) + \sum_{n=1}^{N} \exp\left(f_{sim}(z_q, z_n)/\tau\right)} \tag{2.8}$$

The similarity function f_{sim} gives an estimate of the similarity of two vectors z_1, z_2. The cosine similarity is a popular choice.

$$f_{sim}(z_1, z_2) = cos(z_1, z_2) = \frac{z_1 \cdot z_2}{||z_1||||z_2||} \tag{2.9}$$

2.1.3 Optimizers

In this section, we will give an overview of relevant optimizers that are used to update our models in our papers A to E.

Stochastic Gradient Descent. SGD is the simplest way of updating the model, nevertheless, it is still a widely used optimizer. The network parameters Θ are updated towards the negative direction of the gradient of loss $\nabla_\Theta \mathcal{L}(\Theta)$. The learning rate α regulates the step size during the update. [Bot10]

$$\Theta_{t+1} = \Theta_t - \alpha \nabla_{\Theta_t} \mathcal{L}(\Theta_t) \tag{2.10}$$

Updates can be stabilized with **momentum**, where the weights are updated according to the running average of the gradients of previous iterations. The hyperparameter momentum rate β controls how much the newest gradient affects the running average, i.e. momentum. [GBC16]

$$v_t = \beta v_{t-1} + (1-\beta) \nabla_{\Theta_t} \mathcal{L}(\Theta_t) \tag{2.11}$$

$$\Theta_{t+1} = \Theta_t - \alpha v_t \tag{2.12}$$

AdaGrad. AdaGrad introduces adaptive learning rates. For each weight, local learning rates are determined based on their prior gradient updates. More precisely, the local learning rate is computed by dividing the global learning rate α with the square root of the sum of the squares of all past gradients $g_{t,\theta}$. Note, that θ represents the local weights of the network parameters. [DHS11]

$$g_{t,\theta} = g_{t-1,\theta} + (\nabla_{\theta_t} \mathcal{L}(\theta_t))^2 \tag{2.13}$$

$$\theta_t = \theta_{t-1} + \frac{\alpha}{\sqrt{g_{t,\theta}} + \epsilon} \nabla_{\theta_t} \mathcal{L}(\theta_t) \tag{2.14}$$

Adam. Adam is one of the most popular optimization algorithms because minimal tuning is required for successful training. However, compared to a well-tuned SGD, it converges slower. Two momentums are introduced, one for the gradient update and one for the local learning rate update. The momentum strength can be controlled with the momentum rates β_1 and β_2. [KB15]

$$v_{t,\theta} = \beta_1 v_{t-1} + (1-\beta_1) \nabla_{\theta_t} \mathcal{L}(\theta_t) \tag{2.15}$$

$$g_{t,\theta} = \beta_2 g_{t-1,\theta} + (1-\beta_2)(\nabla_{\theta_t} \mathcal{L}(\theta_t))^2 \tag{2.16}$$

$$\theta_t = \theta_{t-1} + \frac{\alpha}{\sqrt{g_{t,\theta}} + \epsilon} v_{t,\theta} \tag{2.17}$$

AdamW. The AdamW is a popular variant of Adam and extends Adam with a decoupled weight decay to avoid model overfitting. The strength of the weight decay is controlled with the hyperparameter λ. [LH17]

$$\theta_t = \theta_{t-1} + \frac{\alpha}{\sqrt{g_{t,\theta}} + \epsilon} v_{t,\theta} - \alpha \lambda \theta_t \tag{2.18}$$

Ranger. The list of incremental improvements for the Adam optimizer continues and is large, therefore, out of scope for this thesis project. However, Wright et al. [WD21] carefully reviewed and tested a number of potential improvements and suggest a combination of AdamW with eight additional improvements, which they call Ranger. For the training of ResNet-50 on ImageNet, the Ranger optimizer accelerates training and achieves higher validation performance. [WD21]

2.1.4 Regularization methods

Overfitting describes the effect that occurs during model training when the model is able to adequately classify the training samples but performs poorly on samples beyond that. However, it is desired to train a model that generalizes over the training data. Different regularization techniques have been proven to be effective. In the following, we present the most popular ones, that have also been partially used within this thesis project.

Weight regularization. The loss function can be extended with an additional term, to regularize the network weights.

$$\mathcal{L}(\Theta)_R = \mathcal{L}(\Theta) + \lambda_R \Omega(\Theta) \tag{2.19}$$

The most frequently used techniques are L1 and L2 regularization. The first adds the sum of absolute weights, while the latter adds the sum of squared weights. While L2 regularization aims to keep all weights small, L1 regularization is linear and, therefore, allows for large weights when at the same time counterbalancing weights are going toward zero. [GBC16]

Dropout. During training, individual neurons are deactivated with a probability of p. The active neurons are forced to compensate for the missing neuron capabilities of the dropped-out ones. This prevents the network from being dependent on single neurons, leading to improved generalization capabilities. [Sri+14]

Batch normalization. For each layer, the input is normalized over the mini-batch so that the values are zero-centered with a standard deviation of one. Batch normalization removes the necessity of learning the underlying input distribution, which changes along with the training progress. Ioffe and Szegedy [IS15] show a speedup in training convergence as well as an improved network generalizability.

Data augmentation. The more diverse data we feed the network with, the better it can generalize over unseen datapoints. In most cases the number of datapoints is limited, however, augmentation techniques can enlarge the data information. In the domain of computer vision, this is mainly done with spatial-level image transformation, e.g. image rotation, cropping, scaling, and pixel-level image transformation,

e.g. color jitter, random blur, brightness, and contrast variations. In this project, the augmentation techniques of the library albumentations [Bus+20] have been mainly applied. For contrastive self-supervised learning, *multi-crop* [Car+20] has been proven to be an effective augmentation method, where positive pairs are constructed with crops of the same resolution but also with crops of lower resolution.

2.2 Applications of deep learning in visual perception

One major application of deep learning architectures is the domain of visual perception. Particularly, efficient operations such as convolutional layers, mini-batch training, and the increasing availability of computational resources enable training with large amounts of images at a reasonable duration.

2.2.1 Metrics

Metrics are used to measure the final network performance and they depend heavily on the underlying task. Usually, metrics are computed for the validation and test data split. In the following, we introduce the metrics that have been relevant to the models of this thesis project.

Precision and Recall. Predictions can be divided into four categories.

- True Positives (TP): A correct foreground prediction.
- True Negative (TN): A correct background prediction.
- False Positive (FP): An incorrect foreground prediction. This can be caused by an incorrect class assignment or an imprecise object localization.
- False Negative (FN): An incorrect background prediction, because the ground truth is represented by a foreground object.

Precision P describes the percentage of correct predictions. Recall R describes the percentage of correct predictions among all ground truth foreground objects.

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \tag{2.20}$$

Accuracy. The accuracy describes the number of correct predictions over the number of all predictions.

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.21}$$

F1 score. Recall and Precision are combined in a single metric. [CL13]

$$F1 = \frac{2RP}{R+P} = \frac{TP}{TP + 0.5(FP + FN)} \tag{2.22}$$

Jaccard Index. The Jaccard Index [Jac12] – also referred to as Intersection over Union (IoU) – describes the localization accuracy and computes the degree of overlap between prediction $\hat{Y}$ and ground truth Y. It can be calculated for bounding boxes as well as for segmentation masks.

$$JC = \frac{|Y \cap \hat{Y}|}{|Y \cup \hat{Y}|} \tag{2.23}$$

Boundary F1 score. The boundary F1 score evaluates segmentation masks based on their contours instead of their regions. For each class, the contour map is generated for the ground truth Y_B and prediction $\hat{Y}_B$. The precision and recall are computed from those contour maps, while a pixel is a TP if the distance between the contours is below error e_B which is equal to 0.75% of the image diagonal. [CL13]

$$P_B = \frac{1}{|\hat{Y}_B|} \sum_{z \in \hat{Y}_B} [[d(z, Y_B) < e_B]] \qquad R_B = \frac{1}{|Y_B|} \sum_{z \in \hat{Y}_B} [[d(z, Y_B) < e_B]] \tag{2.24}$$

The Boundary Precision P_B and Recall R_B are then used to compute the F1 score as presented in Equation 2.22.

Average Precision (AP). The Average Precision is the area under the precision-recall curve. To construct the precision-recall curve, all predictions are sorted by confidence, where Precision and Recall are computed for the accumulative predictions. These Precision-Recall pairs are plotted against each other, resulting in the precision-recall curve, whereas the area under this curve represents the AP.

mean Average Precision (mAP). The mean Average Precision (mAP) averages the AP over the number of classes. In the domain of object detection, a bounding box detection counts as TP if the predicted bounding box overlaps with the ground truth bounding box by a pre-defined threshold, such as $IoU > IoU_{thresh}$. We consider two variants: mAP_{50} and $mAP_{50:95}$. mAP_{50} represents a mAP at $IoU_{thresh} = 0.5$, while $mAP_{50:95}$ additionally averages the mAP over ten different IoU thresholds from 0.5 to 0.95 with an interval of 0.05. In $mAP_{50:95}$ predictions of more precise overlap (i.e. higher IoU) have a stronger impact on the final score[1].

[1] https://cocodataset.org/#detection-eval

The complex nature of mAP makes it difficult to interpret specific model characteristics. Depending on the targeted real-world application, some detection errors are less serious than others. For example, tumor detection aims to minimize the number of missing detections. Incorrect detections as well as localization errors do not matter so drastically, because they can be corrected manually by an expert when checking the discovery. On the contrary, for the application of weed detection, it is desired to reduce the number of incorrect detections, because removing other, potentially good plants wastes resources and prolongs the working time of the robot. Missing detections are not so problematic, as long as the majority of weeds are removed successfully. To get more in-depth information about the model performance characteristics, Bolya et al. [Bol+20] propose an analysis of the sources of error as listed in the following.

- Classification Error ΔmAP_{Class}: The prediction has been correctly localized, but the detection has been assigned to the incorrect class.
- Localization Error ΔmAP_{Loc}: The prediction can be assigned to a ground-truth bounding box, but the IoU falls below the IoU_{thresh}.
- Duplicate Detection Error ΔmAP_{Dupl}: The prediction can be assigned to a ground-truth bounding box with an $IoU \geq IoU_{thresh}$, but there is already another higher-scoring prediction that was assigned to the same ground-truth bounding box.
- Background Error ΔmAP_{Bkg}: A prediction that does not overlap with any ground-truth bounding box.
- Missed Ground Truth Error ΔmAP_{Miss}: Undetected ground-truth bounding box, that do not fall under localization error.

Object Keypoint Similarity (OKS). OKS gives a measure of keypoint predictions. For each keypoint, the Euclidean distance d_i is computed between prediction and target. d_i is normalized by the scale s which corresponds to the object size as well as σ, which represents the standard deviation between human annotations and true labels. In our experiments, we fix σ to a constant value of $\sigma = 0.05$ for all keypoints. To bring OKS in the range of $[0, 1]$, the exponential of the normalized d_i is taken[2].

$$OKS = exp\left(-\frac{d_i^2}{2s^2\sigma_i^2}\right) \tag{2.25}$$

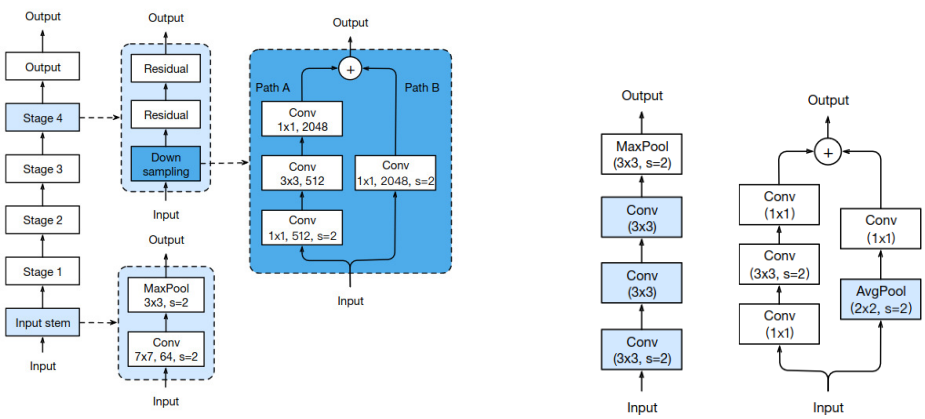
2.2.2 Backbone networks

Backbone networks serve as feature extractors, mapping the input image from high-dimensional space to latent space. The latent space variable is a compressed representation of the image. Backbone networks are used in various downstream tasks such as

[2] https://cocodataset.org/#keypoints-eval

image classification, object detection, object segmentation, keypoint detection, and self-supervised learning. Their capabilities are crucial for the performance of downstream tasks that build on top of it.

ResNet. The first well-known and widely used backbones were AlexNet [KSH12], VGG [SZ14] and Inception [SIV16]. They followed the strategy, that deeper models with a large amount of layers lead to better classification performance. However, deeper models are harder to optimize, because they can easily be caught in a local minimum. ResNet [He+16a] introduced residual layers, which drastically simplified the optimization process. Stacks of convolutional layers are equipped with skip connections which add the identity input directly to the output of the layer stack. The different ResNet variants – with 18, 34, 50, 101 layers – are still a very popular backbone choice today. In Figure 2.1a, the architecture of ResNet-50 is presented. It consists of an input stem with a convolutional layer of 64 (7×7) filters, which are applied with a stride of two, followed by a (3×3) max-pooling layer. The four different stages consist of repeating residual blocks, while the first block has downsampling properties by applying the first convolutional layer with a stride of two. In order to match the dimension of the skip connection, in path B, a (1×1) convolutional layer with a stride of two is applied to the identity input.



(a) ResNet-50 architecture **(b)** xResNet modifications

Figure 2.1: (a) The basic ResNet-50 architecture is illustrated. (b) xResNet applies small modifications to the original ResNet architecture at the input stem (left) and the downsampling residual blocks (right) of each stage. The figures are taken from He et al. [He+19].

xResNet. A modified variant xResNet has been proposed by He et al. [He+19]. Note, that the name xResNet was not originally given by its authors [He+19], but was established by the FastAI community[3]. Small modifications have been applied at the input stem (Figure 2.1b left) and downsampling block (Figure 2.1b right). The (7×7) convolution in the input stem of the ResNet is replaced with three (3×3) convolutions. It is increasing the network's efficiency because a (7×7) convolution is 5.4 times more expensive than a (3×3) convolution. Furthermore, in the downsampling block, for both paths, modifications are applied: in path A, the stride is applied in the second convolution instead of the first, and in path B, a (2×2) average pooling with stride 2 precedes the convolutional layer, whose stride is reduced to 1.

DarkNet-53. The backbone DarkNet-53 has been released as part of the popular object detection network YOLOv3 [RF18]. It consists of 53 layers and includes residual blocks, inspired by He et al. [He+16a]. However, compared to Resnet variants of similar size (101 and 152), DarkNet-53 is more efficient, resulting in higher inference speed [RF18]. Originally DarkNet-53 was implemented in the framework *Darknet*[4], which did not grew as big compared to *tensorflow*[5] and *pytorch*[6]. Although today, DarkNet-53 implementations are available in the more popular deep learning frameworks, the backbone is mainly used in combination with YOLO-based architectures but not in other downstream tasks.

HRNet. Sun and Zhao et al. [Wan+21a] propose a network architecture where connections between high- and low-resolution convolutions are established. The network consists of multiple tracks of different resolutions. In between the tracks, information is exchanged and merged. The main advantage of this architecture is that high-resolution information is maintained through interconnections, which is particularly relevant for dense prediction downstream tasks.

2.2.3 Object detection approaches

Object detection architectures aim to localize and classify objects in images. The most common format of localizing the objects are bounding boxes which enclose the object with a rectangle.

YOLOv5. The YOLO family is a one-stage object detection model achieving real-time performance. YOLOv5 [Gle22] is the first variant that has been natively implemented in the *pytorch* framework, gaining popularity in this large community. It attracted additional contributors which resulted in continuous enhancements from its initial release. Although the major design choices are inspired by the preceding vari-

[3] https://docs.fast.ai/vision.models.xresnet.html
[4] https://pjreddie.com/darknet/
[5] https://www.tensorflow.org/
[6] https://pytorch.org/

ants YOLOv3 [RF18] and YOLOv4 [BWL20], YOLOv5 surpasses the performance in accuracy and inference speed, thus, it is widely applied for various object detection tasks. The network architecture can be divided into three parts.

- Backbone. The residual blocks in DarkNet-53 are extended with Cross Stage Partial (CSP) connections, inspired by Wang et al. [Wan+19a]. The input feature map is divided in half, while one half is processed by the residual block as usual and the other half skips the residual block and is concatenated with the output of the residual block. This design choice leads to a more lightweight architecture – faster training, inference, and less required memory – while keeping the accuracy on par.

- Neck. PANet [Wan+19b] processes the intermediate features from the backbone, while the residual blocks of PANet are again equipped with CSP connections. The neck outputs three feature maps on different resolutions, i.e. (80×80), (40×40), (20×20).

- Head. On top of each feature map, a (1×1) convolutional layer is applied. For each grid position and anchor box, the head predicts whether an object of interest is present. Anchor boxes are predefined bounding boxes, which are refined to enclose the object more precisely. Hence, for each detected object, anchor box refinements as well as class probabilities are provided by the head.

YOLOX. YOLOX builds upon the YOLOv3 variant and adopts it to newer state-of-the-art object detection advancements. Extracted features are processed by a decoupled head, while one pathway is dedicated to the classification and the other is dedicated to the localization of the object. Bounding box predictions are not anymore made based on anchor boxes, but the box parameters are predicted directly (top-left corner, width, and height). While in YOLOv3 only one ground-truth bounding box is considered during loss computation, YOLOX takes into account multiple ground-truth bounding boxes, which fall within a 3×3 center area of the prediction, inspired by [Tia+19]. A more advanced label assignment strategy SimOTA [Ge+21] is applied, which selects the k-top positive labels based on a defined cost function. All remaining labels are treated as negative.

CenterNet. The idea of Centernet [ZWK19] is to identify object instances as keypoints and regress additional properties, such as bounding box dimensions wrt. to the center point. It is an anchor-free way of identifying objects. For more detail see subsection 2.2.5. CenterNet has been proven to be a solid and robust object detection approach while being easily extendable to new custom regression tasks like e.g. pose estimation [ZWK19], instance segmentation [Wan+20] or multi-object tracking [Zha+21].

2.2.4 Image segmentation approaches

Image segmentation architectures aim to make predictions on pixel-level. The literature differentiates between three segmentation tasks: semantic segmentation, instance segmentation and panoptic segmentation. In semantic segmentation, each pixel is assigned to a class, which can represent objects (e.g. cars, persons, plants) or uncountable stuff (e.g. sky, soil). Instance segmentation only targets countable objects and assigns a class and an instance id to pixels, differentiating between instances of the same class. Finally, panoptic segmentation combines the world of semantic and instance segmentation and assigns semantic labels for objects and stuff as well as instance ids for the different object instances.

UNet. The UNet [RPB15] is a fully-convolutional architecture with an encoder-decoder structure. The encoder network can be any convolutional backbone, mapping the image to latent space. The decoder network mirrors the structure of the encoder by replacing the down-sampling operation with an up-sampling operation. The latent space representation is mapped back to the same spatial dimensions as the input image, representing the predicted segmentation mask. Furthermore, Ronneberger et al. [RPB15] introduced skip-connections, which concatenate additional information from encoder to decoder in each spatial stage. UNet is a relatively old architecture from 2015, but it remains a popular choice due to its simplicity.

PSPNet. PSPNet [Zha+17] builds upon a classic encoder-decoder structure while extending the bottleneck with a *Pyramid Pooling Module*. The input feature map at the bottleneck is pooled at different sizes to capture local and global context information. Each pooled representation is again passed through a convolutional layer and an upsampling layer to ensure the same feature dimensions. These upsampled feature maps as well as the original feature map are concatenated and processed further by the decoder component.

2.2.5 Keypoint detection approaches

Keypoint detection aims to identify distinct interest points within an image. Most keypoint detection architectures are evaluated on datasets targeting human pose estimation, where joints of the human body serve as interest points, like e.g. the left/right knee or lef/right shoulder.

CenterNet. The overall idea of CenterNet [ZWK19] is to identify object instances as keypoints. An encoder-decoder network structure is used to produce a heatmap, in which each peak in the heatmap represents a detected object. Additional properties are regressed wrt. the center point position, such as e.g. the keypoint positions relative to the center point. Hereby, image features at the center point position are considered for the regression. Thus, it can be infeasible to regress object properties that depend on features deviating spatially from the center point position. This especially

applies to large and/or elongated objects. Deformable convolutions (DC) [Zhu+18] are incorporated in the model architecture to increase the receptive field and allow the model to learn features spatially distant from the center point position.

YOLO-Pose. Similar to CenterNet [ZWK19], YOLO-Pose [Maj+22] extends each *detection head* in the YOLO architecture with an additional *pose head*, which regresses keypoints wrt. the bounding box center. They integrate the *pose head* add-on in YOLOv5, and the anchor-free variant YOLOX.

2.2.6 Contrastive self-supervised learning

Recently, self-supervised learning using Noise Contrastive Estimation (NCE) [GH10; OLV18] has gained attention, as it provides results comparable to state-of-the-art fully supervised methods trained on ImageNet [Den+09]. In 2020 and 2021, multiple highly relevant approaches were published, namely SimCLR [Che+20], Moco [He+20], BYOL [Gri+20], SwAV [Car+20], SimSiam [CH21], BarlowTwins [Zbo+21].

SwAV. In paper B, we used SwAV [Car+20] to obtain better domain-specific pre-trained weights. It combines cluster-based learning approaches with contrastive learning. Figure B.3 illustrates the architecture of SwAV. A batch X of images $x_1, ..., x_n$ is augmented twice to obtain batches X_1, X_2. Both augmented batches are processed by the same encoder network to obtain corresponding feature vectors Z_1, Z_2. These features are assigned to clusters Q_1, Q_2 by mapping Z_1, Z_2 to k learnable prototypes, which are further processed by the Sinkhorn Knopp algorithm [Cut13]. For the computation of the loss, as shown in Equations B.1 - B.3, let's assume one image x_i from X with two augmented views x_{i1}, x_{i2}. The obtained cluster assignments q_{i1}, q_{i2} serve as labels for the images. The trick hereby is to swap the assigned clusters between the image views, i.e. q_{i1} serves as a target for x_{i2} and vice-versa (Equation B.1). Concretely, the loss function is a sum of cross-entropy losses computed for the predictions z_{i1}, z_{i2} and the swapped targets q_{i2}, q_{i1} (Equations B.1, B.2). The probability of the predictions is computed by taking the Softmax of the vector similarity between the feature vector z_i and the prototype vectors C (Equation B.3).

$$L(z_{i1}, z_{i2}) = l(z_{i1}, q_{i2}) + l(z_{i2}, q_{i1}) \tag{2.26}$$

$$l(z_i, q_i) = -\sum_k q_i^{(k)} \log p_i^{(k)} \tag{2.27}$$

$$\text{where } p_i^{(k)} = \frac{\exp(z_i^T c_k/\tau)}{\sum_{k'} \exp(z_i^T c_{k'}/\tau)} \tag{2.28}$$

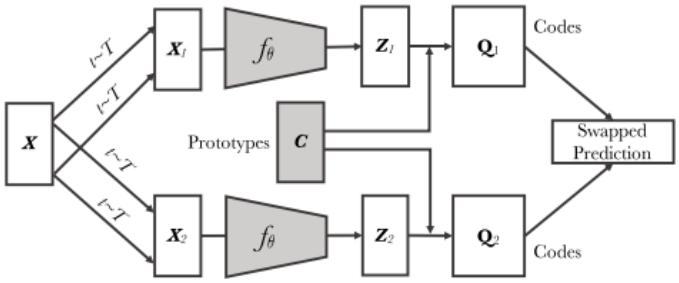


Figure 2.2: Architecture of SwAV [Car+20]: An image batch X is augmented twice to obtain (X_1, X_2) which are then processed by the identical feature extractor f_θ. The obtained feature vectors (Z_1, Z_2) are mapped to prototype vectors that are further clustered to (Q_1, Q_2). The clusters serve as pseudo labels and are swapped between the two streams so that the loss is computed between Z_1 and Q_2 and vice versa.

The main advantage of SwAV is that it allows negative and positive examples to be similar, which can be easily the case when two different images are semantically similar. Furthermore, SwAV computes cluster assignments online and works with relatively small batch sizes, which makes it also applicable with limited available computational resources.

CHAPTER 3

Related Work

Robotic perception in agriculture is a huge field with tasks varying from weed management [Wey+22b; GEN23; Ols+19] to harvesting [Par+; Bir+20], seeding/planting [Ble+16], and monitoring [Cub+20]. All of the previously mentioned tasks can be applied in greenhouses with controlled environmental conditions or in outdoor fields with varying environmental influences.

We will predominantly focus on related work in the area of weed management in outdoor fields. Although our targeted fields are within the grassland domain (i.e. dairy farming), we also refer to crop fields, which are more intensively researched and their work can often be related to grassland fields. However, in crop fields, plants are clearly separable from the soil by means of simple color thresholding or by considering near-infrared (NIR) imagery that has higher reflectivity on chlorophyll content. This is not the case for grassland fields, where foreground vegetation (weeds) and background (grass) are less distinctive because they share similar color and chlorophyll content. Another challenge is, that foreground plants are often partially occluded by the background plants, which makes the prediction task more difficult.

3.1 Agricultural weed datasets

There exists a large number of different agricultural datasets. For this work, we consider grassland field datasets and crop field datasets as relevant, which will be addressed in the corresponding subsections. Finally, in subsection 3.1.3, we present the few existing agricultural robotics datasets which include image data accompanied by measurements from the other robot sensors.

3.1.1 Grassland field datasets

In the domain of grassland agriculture, only four relevant and publicly available datasets were identified. They are shortly summarized in the following.

The *Rumex Ancenis dataset* [KTN19] contains grassland images with *Rumex obtusifolius L.* bounding box annotations. However, the dataset consists only of images taken within one field and one day as well as under controlled lighting conditions. The variety of the image data is therefore low.

Valente et al. [Val+22] provide a dataset consisting of orthophotos, which were recorded with an UAV flying at three different heights (10, 15, 30m). The images were collected within one day in April 2018 and are not suitable for the detection of small plants. For the images taken at 10 and 15m height, experts were able to identify only 80% of the Rumex weeds in the field, and for images taken at 30m height, no Rumex weeds could be identified.

The *GrassClover* dataset [Sko+19] targets the segmentation of clover and grass in order to estimate the biomass composition on grassland fields. The images are generated synthetically using image composition, i.e. no extensive manual pixel-wise annotations were performed. Therefore, the images are not fully representative of real weeds.

Finally, the *DeepWeeds* dataset [Ols+19] is a large image classification dataset including eight common Australian weeds in their natural environment. The background is not strictly green grass due to the dry region, however, rangeland is specifically targeted and can therefore be assigned to the same category.

3.1.2 Crop field datasets

For outdoor crop fields, we can identify a large number of image datasets, targeting different types of crops (and accompanying weeds) such as cauliflowers [BWS19; Kie+23], brocolli [BWS19], sunflowers [Faw+21], potatos [Hin+23], wheats [Dav+21], carrots [Lam+17; HO15; Bos+20], onions [Bos+20] and sugar beets [Wey+23; Che+17; Sa+18]. Previously mentioned datasets provide different characteristics like the number of images per dataset, the annotation type (bounding box instances, semantic segmentation, plant instance segmentation, leaf instance segmentation, joint stem annotation), and the image type (RGB, stereo, multi-spectral, thermal), as shown in Table 3.1.

The majority of previously mentioned datasets are fairly small in size and have been generated to solve specific tasks. However, in the agricultural domain, we still lack benchmark datasets like in other vision domains, e.g. Coco [Lin+14] for general-purpose images and CityScapes [Cor+16] for traffic images. Benchmark datasets are heavily used by the research community to show the performance of newly proposed approaches. Since everyone validates on the same benchmark dataset, results are easily comparable to each other. Two recently published agricultural datasets [Wey+23; Ste+23] try to fill this gap. It remains to be seen whether the datasets will become established and accepted as a benchmark by the community.

PhenoBench [Wey+23] contains 2872 labeled and 1000 unlabeled images of sugar beet fields. They provide different annotation types: semantic masks for the classes soil, crop, and weed; instance masks for all plants; and leaf instance masks for all crop plants. Additionally, they created different benchmark tasks, such as semantic, instance, and panoptic segmentation as well as detection tasks with bounding boxes. To promote the dataset as a benchmark, they organized a challenge at the Inter-

Table 3.1: Overview of the existing crop field datasets with the different target crops, number of annotated images, annotation types (bounding box, semantic segmentation, plant instance segmentation, leaf instance segmentation, joint stem annotation) and image types (RGB, multi-spectral, thermal and stereo).

Paper	target crop	#images	annotation type	image type
Haug et al. [HO15]	carrot	60	sem. seg.	RGB
Lameski et al. [Lam+17]	carrot	39	sem. seg.	RGB
Chebrolu et al. [Che+17]	sugar beet	280	sem. seg.	RGB stereo
Sa et al. [Sa+18]	sugar beet	1670	sem. seg.	RGB multi-spectral
Bender et al. [BWS19]	cauliflower broccoli	1248	bbox sem. seg.	RGB stereo multi-spectral thermal
Bosilj et al. [Bos+20]	carrot, onion	40	sem. seg.	RGB multi-spectral
Fawakherkji et al. [Faw+21]	sunflower	500	sem. seg.	RGB multi-spectral
David et al. [Dav+21]	wheat	6422	bbox	RGB
Kierdorf et al. [Kie+23]	cauliflower	2198	sem. seg. plant inst. seg. leaf inst. seg. joint stem	RGB multi-spectral
Weyler et al. [Wey+23]	sugar beet	2872	sem. seg. plant inst. seg. leaf inst. seg.	RGB
Hindel et al. [Hin+23]	potato	1433	bbox	RGB multi-spectral
Steiniger et al. [Ste+23]	16 crop species	8034	bbox plant inst. seg. joint stem	RGB

national Conference for Computer Vision (ICCV) in 2023 to perform *Hierarchical Panoptic Segmentation* (i.e. semantic and instance segmentation). We successfully participated in the proposed challenge and ranked 3rd place from 25 participants in total.

The *CropAndWeed* dataset [Ste+23] contains 8.034 labeled images with 24 labeled plant classes including 8 crop labels and 16 weed labels. Images were collected on a variety of different fields — real fields and cultivated fields with rare species. They emphasize the variety of their crops and weeds in the early growth stages. They provide annotations of different types such as bounding boxes, joint stem positions, and segmentation masks for each instance as well as metadata about the environmental conditions like e.g. the degree of moisture, and lightning conditions.

3.1.3 Agricultural robotics datasets

The generation of a sole image dataset requires significantly less effort than the generation of a robotics dataset, where not only images but also data from additional robot sensors, such as GPS, IMU, odometry, or LIDAR, are collected. A robotics dataset must be collected with an actual robot platform and the data management complexity is notably higher. Obviously, these datasets are richer in information and allow one to relate datapoints from different sensors, resulting in different possible applications, e.g. visual SLAM and object mapping. In the following, we summarize three relevant robotics datasets within agricultural field robotics—even though, none of them targets grasslands.

The *SugarBeet* dataset [Che+17] was the first agricultural field robotics dataset collected with the *BoniRob* platform as introduced in [Che+17]. Data was recorded over three months in a sugar beet farm, including a high number of different sensors: RGB camera, stereo camera, laser scanner, GPS, and odometry. 280 images are manually labeled with pixel-wise segmentation masks, including the sugar beet crop and nine different types of weed.

For the *Brassica* dataset [BWS20], weekly sequences of cauliflower and broccoli were collected to capture the growth cycle. Their robot platform *Ladybird* is equipped with a GPS, IMU, a forward-pointing line-scan hyper-spectral camera, a downwards-pointing stereo camera in controlled lighting conditions as well as a downwards-pointing thermal camera. For a selected subset of the acquired images, ground-truth annotations on pixel– as well as bounding-box–level are provided.

Finally, the *Rosario* dataset [Pir+19] is a multi-sensor dataset—stereo camera, IMU, GPS, odometry—targeting specifically the application of visual SLAM. The stereo camera is pointing towards the forward-driving direction and no image annotations are provided.

In paper C, we provide the first robotics dataset for grassland fields. It has a decent size with 5510 images constituting a candidate for an agricultural benchmark dataset in grasslands.

3.2 Representation learning in the domain of agriculture

In 2020, self-supervised learning regained great attention due to the successes of contrastive approaches [Che+20; He+20; Gri+20; CH21; Car+20], which show comparable results to state-of-the-art supervised methods trained on ImageNet [Den+09].

On the contrary, in the domain of agriculture, self-supervised learning has not been seriously explored to that point in time. Common practice was to use ImageNet weights for transfer learning, because many publications in the domain of agriculture, e.g. [KTN19; MYA17; Suh+18], show that ImageNet pre-trained weights improve the model performance. Little research has been performed on whether more domain-specific transfer learning could improve performance even further.

Bargoti and Underwood [BU17] use transfer learning to benefit from weights in-between fruit orchards. Transferring weights from other orchards gives only advantage over ImageNet weights when training on a limited set of images.

Bosilj et al. [Bos+20] train supervised on one specific crop and transfer these weights to fine-tune on another crop. They show that initializing networks with similar-domain weights leads to more efficient training and only requires a fraction of the ground truth labels to achieve similar performance.

Espejo-Garcia et al. [Esp+20a] generate agricultural pre-trained weights on the highly diverse, but relatively small *Plant Seedlings Dataset* [MYA17] and test them on the more specific *Early Crop Weeds Dataset* [Esp+20b]. They perform experiments for four popular CNN architectures and report an overall slight improvement in performance and a reduction in training convergence time.

Marin Zapata et al. [Mar+20] learn a representation of a plant phenotyping dataset with a triplet network architecture, inspired by Wang et al. [Wan+14]. They take advantage of the underlying timing sequence of the images in their dataset, where consecutive image frames are treated as similar and non-consecutive image frames as dissimilar. The learned representation is used in various downstream tasks such as clustering and classification.

Zhao et al. [Zha+20b] propose a multi-task learning framework that combines supervised classification with a hand-crafted self-supervision task, inspired by Gidaris et al. [GSK18]. The purpose of the self-supervision component is to speed up the optimization by setting a focus on the encoding of image orientation.

With our paper B from 2021, to the best of our knowledge, we were the first to explore self-supervised contrastive learning for the generation of agriculture-specific pre-trained weights. We obtained weights from three different, relatively large datasets, which were evaluated on classification and segmentation downstream tasks.

Thereafter, numerous publications that apply self-supervised learning in agriculture have been released. Within the scope of this thesis, we will present the ones related

to weed and crop management in outdoor fields.

Ogidi et al. [OES23] use two popular SSL approaches MoCov2 [He+20] and DenseCL [Wan+21b] to generate pre-trained weights on 4 different pre-training datasets (ImageNet-1k [Den+09], 2 dataset variants with images coming from iNaturalist [Van+21], field crop images from Beck et al. [Bec+21]). They evaluate the generated pre-trained weights on 4 different downstream tasks (wheat head detection, plant instance detection, wheat spikelet counting, and leaf counting). All downstream tasks are beyond classification, and therefore, pixel-level features are expected to be relevant. During their experimental evaluation, they show that supervised training outperforms the SSL variants for most downstream tasks. These findings are in accordance with the results in our paper B for the dense prediction downstream task.

Roggiolani et al. [Rog+23b] use Barlow Twins [Zbo+21] to learn pre-trained weights in a self-supervised manner. They pre-train on the images of the target domain and evaluate these weights on two pixel-level downstream tasks, namely semantic segmentation and leaf instance segmentation. They carefully tune the set of augmentations that is used to generate the image views x_1, x_2, serving as input to the Barlow Twins. The usage of SSL weights outperforms over the supervised variant in dense prediction tasks, which was not achieved by our work in paper B.

Hindel et al. [Hin+23] is proposing a new SSL approach to improve particularly on dense prediction tasks, which has been shown to be difficult by us in paper B and Ogidi et al. [OES23]. They inject features from a disjoint *noise dataset* to the feature maps of the *targeted dataset*. Hereby, only features from local regions are injected, and these local regions furthermore serve as pseudo labels for different dense prediction downstream tasks, e.g. an enclosing bounding box of the local region for object detection downstream task or the mask of the local region for a segmentation downstream task. Their proposed approach outperforms both supervised training as well as SSL alternatives such as Mocov2 [He+20], BYOL [Gri+20], InsLoc [Yan+21], DenseCL [Wan+21b] and the work from Roggiolani et al. [Rog+23b].

3.3 Weed detection

Weed detection is a highly researched field and it is common practice to apply well-performing deep vision models to specific agricultural tasks. It ranges from anchor-based detection models like YOLO [Car+23; Soz+22], SSD [Ola+21; Vee+20]; to encoder-decoder models like CenterNet [JCC21; Jin+22], PSPNet [YLL20; Gül+21], UNet [AB20; AA20]; to transformer-based models [Esp+23; JAL23]. Using out-of-the-box implementations of popular network architectures has the great advantage that relevant pre-trained weights are generally available to train efficiently and achieve better results while having limited data available.

The automated detection of the targeted weed of this Ph.D. project *Rumex obtusifolius L.* (broad-leaved dock) has been continuously investigated in agricultural robotics since the early 2000s. It underlines the application's relevance and the need for re-

moving Rumex weed efficiently and robustly at dairy farms. The applied approaches in that field can be assigned to three areas: classic computer vision, machine learning with hand-crafted feature extraction, and deep learning.

Classic computer vision. Researchers from Wageningen Research [Pol+07; Eve+09] applied Fourier Analysis followed by thresholding, assuming that Rumex leaves have lower spectral power than grass leaves. Dürr et al. [Dür+08] use local binary pattern as a feature extractor and classify Rumex weeds by applying pixel-level thresholding. Seatovic and Anken is the only research group so far, that worked with 3D sensor data by applying 3D surface analysis to find Rumex plants with significantly higher leaf surfaces [SWS08; Ank+10].

Machine learning. There have been several studies on combining different hand-crafted feature extractors (e.g. Local Binary Pattern, Pyramid Histogram Of Visual Words, Difference of Gaussian, determinant of the Hessian detector) with non-linear and linear Support Vector Machines (SVM) as a final classifier [Ahm+14; KTN16; BCF18; KTN18; Zha+18].

Deep learning. Following the trend in the last decades, Rumex weeds have been identified using deep learning architectures. Initial attempts have been made to use image classifiers in a sliding window fashion, like e.g. AlexNet [Zha+18; Val+19] and VGG16 [Lam+21]. Kounalakis et al. [KTN19] keep their SVM classifier from their previous work, but use Deep Neural Networks and transfer learning techniques for automatic feature extraction. Valente et al. [Val+22] compare the performance of more advanced models such as Resnet and DenseNet as well as computationally efficient models such as ShuffleNet, MobileNet, EfficientNet, MNASNet. Al-Badri et al. [Al-+22] use ensemble learning by combining the output of three different classification networks.

While previously mentioned methods are all based on the sliding window approach, dense prediction models are gaining interest. Schori et al. [SAŠ19] apply an encoder-decoder network to generate semantic segmentation masks. Object detection models with bounding box predictions have been applied, i.e. Yolov4 [SSP22] and YOLOX in paper C. Al-Badri et al. [Hus+23] use a custom variant of an R-CNN by adding ensemble learning and proposing an adaptive non-maximum suppression, complying better with overlapping objects.

3.4 Fine-grained plant analysis

Fine-grained scene understanding enables more advanced robotic applications. For precision agriculture, a more detailed analysis of the plant can advance the automation of labor and physically demanding tasks, such as phenotyping and organic weed management.

3.4.1 Estimation of leaf count, -size and -shape

The automation of phenotyping in real-world fields has recently shown progress in the extraction of three relevant plant traits to monitor the plant's growth state: the leaf count, the leaf shape, and the leaf size.

Teimuri et al. [Tei+18] applies a classification network Inception [Sze+15] with 9 different classes, where each class represents the number of visible leaves per plant which can range from one to nine. The underlying image data contains 19 different plants of different growing states in their natural environment, including occlusions and unfavorable camera angles. Weyler et al. [Wey+21] use CenterNet [Dua+19], where each keypoint represents a leaf instance. Simultaneously, they also detect crops and weeds with bounding boxes as well as plant instances with keypoints.

A more advanced way to count leaves is leaf instance segmentation. It enriches the leaf count information with the leaf size and shape. PhenoRob Research is the only research group, that has performed leaf instance segmentation in real-world settings. Weyler et. al. propose two bottom-up approaches [Wey+22b; Wey+22a], which apply leaf instance segmentation in combination with other more coarse phenotyping tasks, such as plant instance detection and joint stem detection. Roggiolani and Sodano et al. [Rog+23a] suggest a top-down approach to simultaneously identify semantics, plant instances, and leaf instances. Their model is based on ERFNet [Rom+18], while they introduce new skip-connection in-between decoder heads to establish relations between different tasks. More concretely, information is transferred from the semantic decoder to the plant instance decoder, and from the plant instance decoder to the leaf instance decoder.

3.4.2 Joint stem detection

In recent years there has been an increased interest in joint stem detection as the removal process of plant at the source of the plant, where all stems can be traced back to, is most efficient. It applies particularly for plants of early growing state because it is more feasible to identify and treat the joint stem.

Langer et al. [Lan+18] take the detected plants as input and apply geometrical computer vision methods in order to retrieve the joint stems. The parameters of their methods are tweaked to address very specific plants at specific growing stages; therefore, they don't adapt to different cases.

Deep Neural Networks are applied to detect the weeds and crops as well as their joint stem keypoints in a multi-task setting. Lottes et al. [Lot+18] apply an encoder-decoder network to segment plants and predict joint stem positions. On top of a shared encoder, two decoders follow—one for each task. The same authors improve their model robustness by incorporating spatial-temporal information (i.e. consecutive image sequences) to the network [Lot+20]. Unfortunately, Lottes et al. [Lot+18; Lot+20] do not segment the individual plants as instances, which risks multiple treatments of the same plant.

In contrast, Weyler et al. [Wey+22b] use a bottom-up architecture to simultaneously detect joint stems, overall semantics as well as plant and leaf instances. They follow a topological approach, where the identified leaf keypoints are associated with one joint stem keypoint, and crop pixels are associated with one leaf keypoint.

Lac et al. [Zha+20a] propose a two-stage method, where they use YOLOv4 for fast stem detection, followed by a temporal aggregation algorithm to refine the stem predictions. Note, that they predict joint stem positions with bounding boxes instead of keypoints.

Zhang et al. [Zha+20a] propose, similarly to Lottes et al. [Lot+18], an architecture with one shared backbone but two separate decoder paths to generate segmentation masks for segmenting weeds as well as keypoint predictions for joint stems, based on heatmaps. They additionally provide the feature maps from the weed segmentation path to the joint stem prediction path and claim to increase accuracy by doing that. Note, that the architecture only handles input images containing no more than a single plant each.

In paper D, we apply joint stem detection for the first time to grassland weeds.

3.4.3 Vein detection

The extraction of leaf veins is an important property during plant phenotyping. The length and thickness of leaf veins give insights into the plant's growth state and health; the transportation capacity for water, nutrition, inorganic salts, and trace elements can be determined [Kon+21; Pan+22]. Since the manual measurement of leaf veins is a time-consuming and labor-intensive task, it has been of great interest to automate the process. In recent years, automatic leaf vein segmentation has been performed with classic computer vision approaches [Büh+15; ZW10; Li+11; Zhu+20], traditional machine-learning approaches with hand-crafted feature extraction [Sel+17; LH13; SCB21] as well as few deep learning approaches [Liu+22]. In all aforementioned works, experiments are performed on sharp high-resolution (or even microscope) images in laboratory settings with controlled lighting conditions.

In paper E, we aim to perform automated leaf vein extraction in real-world settings, which has not been explicitly addressed so far. While Marks et al. [MMS22] need to apply 3D reconstruction from multiple image views to extract advanced leaf traits, our approach only requires a single 2D image as input to solve the task of automatically extracting detailed plant traits.

CHAPTER 4

Contributions

The contributions of this work are heavily motivated by the underlying application which is the autonomous and herbicide-free removal of the problematic grassland weed *Rumex obtusifolius L.*. The work is funded by the EU Horizon 2020 project GALIRUMI, in which we aim to solve the task of sustainable weed removal with a commercially available outdoor robot platform while exploring two different weed removal techniques: (1) laser weeding and (2) electric weeding. Laser weeding works by removing the leaves of the weed several times over the course of the growing season. When done often enough, this will eventually exhaust the reserves in the tap root [Eve+20]. Electric weeding works by guiding an electric current through the tap root of the weed, which kills or seriously damages the entire root. A summary of the outcome of the GALIRUMI project can be found in paper F.

In section 4.1, we introduce the robot platform that was predominantly used for the development of the visual perception system, which is addressed in the subsequent sections. Section 4.2 is about the dataset creation process. First, we present potential methods to facilitate the data annotation process in paper A and paper B. Secondly, we give an overview of our two published datasets, *RumexWeeds* in paper C and *RumexLeaves* in paper E. In section 4.3, we present the *Rumex* detection model to identify the individual weed plants in the field, developed in paper C. In section 4.4, we present models developed to perform a more fine-grained analysis of the weed plant including the join stem detection in paper D and the detection of individual leaves and their corresponding stem and vein line in paper E.

4.1 The robot platform

The primary robot of the GALIRUMI project is a CEOL robot from Agreenculture[1] and is located in The Netherlands (see Figure 1.2). It is fully integrated with all relevant hardware components such as the localization, vision, and weeding components. To independently develop the visual perception system, in our research group at the Technical University of Denmark, we created a duplicate integration of the vision components on a different base platform as shown in Figure 4.1, namely the Husky robot from Clearpath Robotics[2]. It is an outdoor ground robot of a smaller size,

[1] https://www.agreenculture.net/en/ceol
[2] https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

which is sufficient for the purpose of data collection and real-world *Rumex* detection tests. The visual perception system is mounted in the same way as on the CEOL robot. Furthermore, we added a low-cost GNSS sensor (Navilock NL-8022MU USB 2.0 Multi GNSS receiver) and a IMU sensor (Xsens MTI-630) for the recording of navigational datapoints.

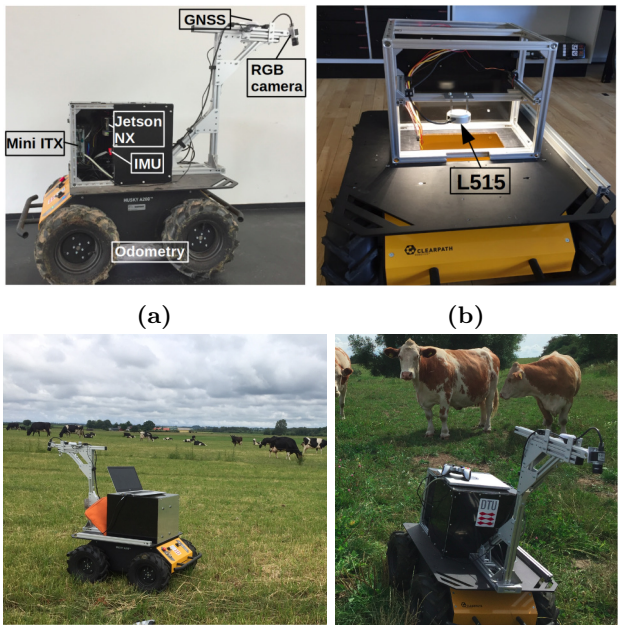


Figure 4.1: We mount the vision components on the Husky robot from Clearpath Robotics. (a) It is equipped with navigational sensors (IMU and GNSS) and two vision cameras: an RGB camera to detect the weed on a broad view and an IntelRealsense L515 Lidar to further analyze the detected weed in a close-up view. (b) The L515 sensor is mounted within a housing with controlled lighting conditions, pointing down through a hole in the robot. (c) We use the duplicate setup for data collection and real-world *Rumex* detection tests.

For the visual perception system, two commercially available cameras are considered: (1) a first-stage camera to be used when the robot is searching for weeds, and (2) a second-stage camera to be used when the robot has come to a stop close to the weed and a close-up analysis of the detected weed must be made. Images from both cameras are processed by a Jetson Xavier NX. The first-stage camera is a Basler ace 2 and mounted at a height of 1m, tilted 75° toward the ground. The setup results in a field coverage of a minimum 1.5 m width. The second-stage camera is an Intel RealSense Lidar Camera L515. It provides RGB as well as depth images of the weed,

that has been initially detected with the first-stage camera. The sensor is mounted 0.5 m above ground pointing straight down to the ground through a hole in the robot. To acquire high-quality depth images, the sensor is mounted in controlled lighting conditions, i.e. under a shield with LED lightning, because the wavelength of natural light interferes with the wavelength of the L515 sensor which reduces the accuracy of the depth map.

4.2 Rumex dataset creation

Deep neural networks require a sufficient amount of training samples in order to show generalisation effects on samples outside the training data. Recent developments have shown the importance of the training dataset size in regards to the model performance. As mentioned in section 3.1, limited real-world grassland image datasets are publicly available. The creation of datasets tackling the grassland weed *Rumex* was inevitable in order to evaluate the perception performance of the GALIRUMI robot.

However, dataset creation is cost-intensive as we aim to collect diverse and representative data, such as different field locations, different years and seasons, as well as different weather conditions (sunny, cloudy, rainy). Furthermore, the manual creation of image annotations is a labor and time-intensive task. To reduce the cost of manual labeling, we have investigated two potential areas: semi-automatic annotation and self-supervised learning.

During data collection, we observed that plants captured within one data collection session (i.e. one field and one day) have relatively low variance in their visual appearance. Plants from the same region have been exposed to similar environmental influences, resulting in similar phenotypic characteristics. In paper A, we build upon that observation and propose a semi-automatic annotation pipeline as illustrated in Figure 4.2. Only a few leaves must be manually annotated to compose new plant arrangements and paste them on a diverse set of pure background images. A segmentation model is trained on the synthetic data and used to generate mask proposals for the real images. Finally, the annotator reviews the mask proposals and optimally, only corrections for difficult out-of-distribution samples are required. At that time in the project, our robot platform wasn't yet available. Therefore, we evaluated the proposed approach from paper A on a limited amount of data, which has been collected with an 8-megapixel 1/3-inch camera while walking on fields. We perform ablation studies for different blending methods during image composition, for the required number of manually labeled leaves as well as for the influence of post-processing mask refinement. Finally, we show that a predominant portion of pixels is predicted with an IoU > 0.6. Later in the project, we integrated the proposed pipeline in our annotation process with CVAT [Sek+20]. Against our initial expectations, the practicality of the pre-annotations using the proposed pipeline proved to be less efficient and accurate than manual annotation. Performing corrections on object proposals is in a majority of cases more time-consuming than annotating the

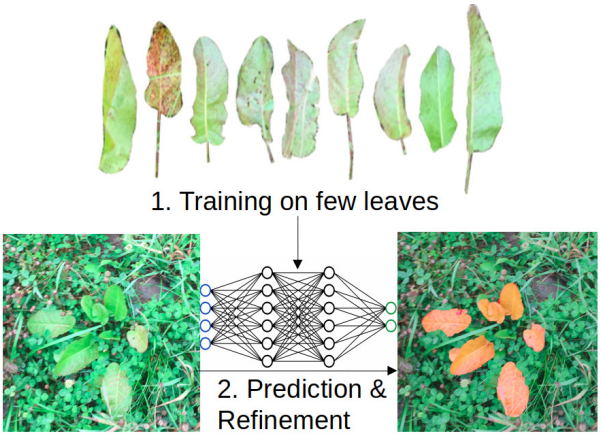


Figure 4.2: *Few-leaf Learning*: A segmentation model can be trained on a very small number of manually annotated leaves composed with pure background images. The figure is taken from paper A.

full image from scratch. Human annotators are less careful when reviewing object proposals and tend to miss required corrections. Furthermore, the approach doesn't work well on occluded objects, which is often the case when the grass has grown to a certain extent and overlaps the *Rumex* weed. In the end, we decided not to use *Few-leaf Learning* for the annotation of our datasets *RumexWeeds* and *RumexLeaves*, which will be introduced at a later point of this section.

Another way of improving label efficiency is the application of transfer learning, which is a popular and effective technique when supervised training is performed with limited data. Established network architectures such as VGG16 [SZ14], ResNet [He+16b], and SwinTransformer [Liu+21] can be initialized with freely available pre-trained weights. Originally, those pre-trained weights were obtained with supervised pre-training on ImageNet [Den+09], which consists of more than 14 million manually annotated natural images. In the domain of agriculture, transfer-learning using ImageNet pre-trained weights has shown to be effective, although ImageNet includes natural images that differ semantically as well as in texture and color from most agriculture image datasets. Can domain-specific pre-trained weights increase the effect of transfer learning? Unfortunately, there exists no agricultural equivalent for ImageNet. However, in recent years, self-supervised training on unlabeled images achieved comparable results. The key idea of self-supervised training is to train the model supervised on images with automatic labels. Automatic labels can be generated in many different ways, e.g. region inpainting, where a random image region is removed from the input image on purpose, and then used as ground truth during loss computation.

Self-supervised contrastive approaches have shown superior results. Automatic labels are generated with image transformations. Image views originating from the

same image provide positive pairs, while samples originating from different images provide negative pairs. Siamese architectures process image pairs and Noise Contrastive Estimation (NCE) loss is used to push positive pairs closer together and negative pairs further apart. In paper B, we apply a state-of-the-art contrastive self-supervised approach - SwAV [Car+20] - to the area of agricultural images. We obtain domain-specific weights for three agricultural datasets: DeepWeeds [Ols+19], Aerial Farmland [Chi+20], and Grassland Europe. The latter consists of 24.000 scraped grassland images. We compare the domain-specific pre-trained weights to ImageNet weights by transferring them to classification and segmentation downstream tasks. Our results show, that self-supervised pre-trained weights are particularly effective in classification downstream tasks in a semi-supervised setting, where only a fraction of annotated images is used. On the condition that the domain-specific dataset is essentially smaller than ImageNet, SwAV allows one to generate domain-specific pre-trained weights for custom architectures with much lower computational cost, compared to weights obtained with ImageNet in a supervised manner. This enables flexibility in network design and doesn't force one to stick to architectures that already come with pre-trained ImageNet weights. Sample results for DeepWeeds [Ols+19] are demonstrated in Figure 4.3.

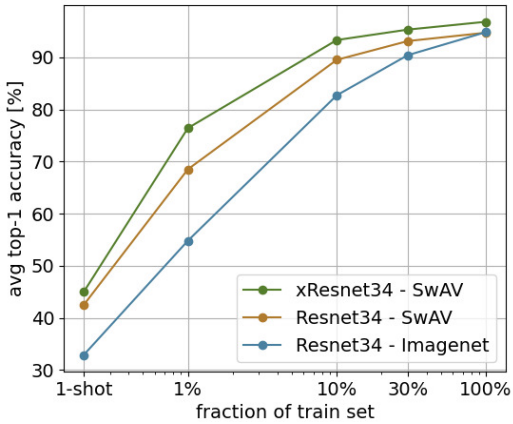


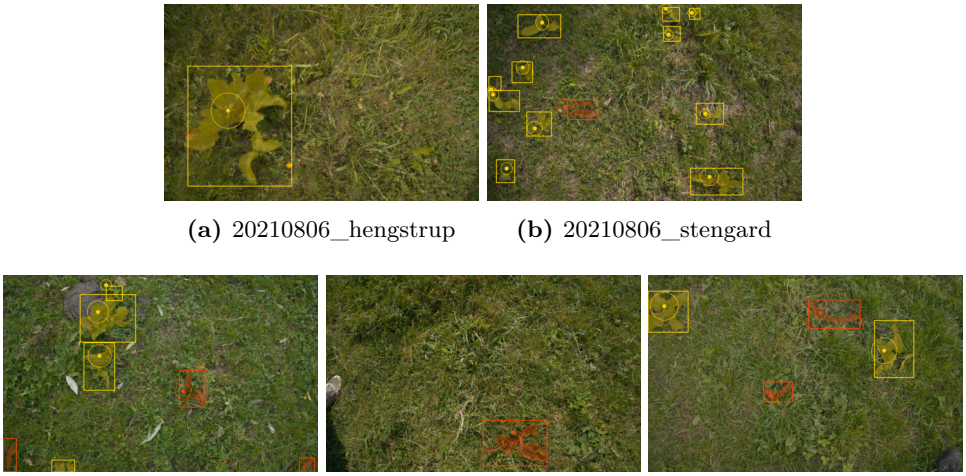
Figure 4.3: Average top-1 classification performance for models trained on different fractions of the DeepWeeds dataset. We apply transfer learning and compare supervised ImageNet training to self-supervised pre-training using SwAV. With an increase of average top-1 accuracy up to 14%, SwAV is particularly efficient in semi-supervised settings, where only a fraction of image annotations are needed. Furthermore, SwAV allows us to easily obtain pre-trained weights for any network structure. Exemplary, we show that minimal network modifications such as concat pooling and xResNet lead to a further performance increase. The figure is taken from paper B.

On the contrary, for dense prediction downstream tasks, SwAV weights are less

effective, showing slightly reduced performance over ImageNet weights. To the best of our knowledge, this study was the first study investigating self-supervised contrastive learning in the domain of agriculture, contributing valuable insights to the community. Within the scope of the GALIRUMI project, we aim to create datasets with dense predictions such as bounding box and segmentation mask annotations. According to the results, it is not beneficial to use SwAV weights over ImageNet weights for dense prediction tasks. Therefore, we solely transferred ImageNet weights in the further course of the thesis.

Within the scope of this thesis, we have created and published two datasets, namely *RumexWeeds* within paper C and *RumexLeaves* within paper E. We give a broad overview of both datasets in the following paragraphs.

The dataset *RumexWeeds* published within paper C can be related to the first stage vision task of the GALIRUMI visual perception system, i.e. the overall detection of *Rumex* plants in the field. For *RumexWeeds*, the robot platform introduced in section 4.1 was used to collect several time sequences with a total of 5,510 datapoints, each containing an RGB image from the first-stage camera as well navigational data from the robot including IMU data, GPS data, and Odometry data. This type of data allows to develop more advanced models by additionally considering the temporal aspect as well the measured movements of the robot. For each image in the sequence, manual bounding box annotations have been performed, distinguishing between two sub-species *Rumex obtusifolius L.* and *Rumex crispus L.*. To support the annotation process, we have developed a custom annotation tool in order to take advantage of the image sequences as well as the navigational robot data. Initial bounding box annotations are tracked using odometry measurements as well as visual CRST [Luk+18]. If both trackers agree, the tracked bounding box is proposed to the annotator. The annotator can accept, modify or delete the proposal. Additionally, pixel-wise annotations have been performed with CVAT [Sek+20] for a randomly sampled subset of images. In total, the dataset provides 15,519 bounding box and 340 segmentation mask annotations. Paper D extends the dataset *RumexWeeds* with manual joint-stem annotations. As the joint stem can be difficult to identify in some images, each annotation is represented with a center point for the approximate joint-stem position and a radius indicating the annotators' confidence in the performed annotation. In Figure 4.4, we show example images of the dataset for different locations and collection days (i.e. <collection day>_<location>). Please refer to paper C for a detailed overview of the datapoints as well as an in-depth dataset analysis.



(a) 20210806_hengstrup **(b)** 20210806_stengard

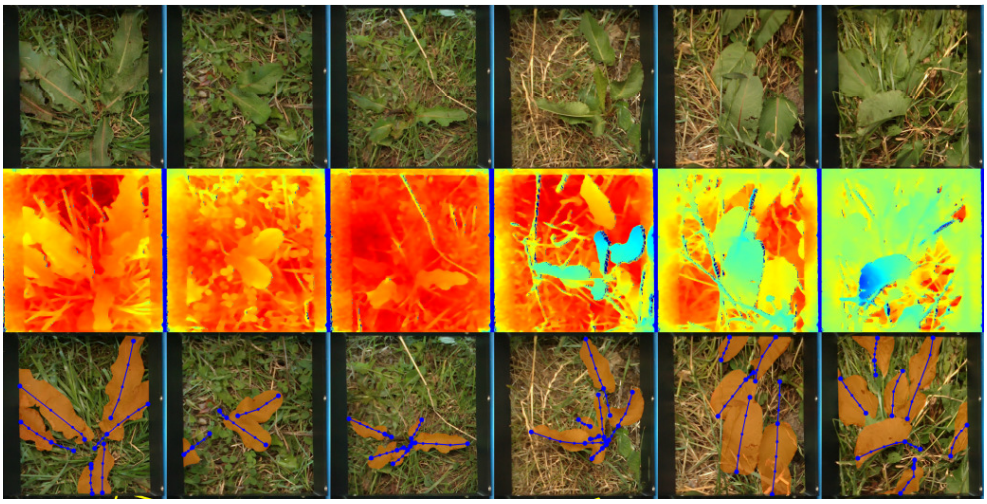
(c) 20211006_stengard **(d)** 20210807_lundholm **(e)** 20210908_lundholm

Figure 4.4: Sample images from the dataset *RumexWeeds* with bounding box, segmentation, and joint stem annotations for each plant. We show one sample image for each data collection session with the ID <collection day>_<location>. A selection of the *RumexWeeds* dataset variety is presented here: sunny vs. cloudy, plant size variation, plant density, and negative objects such as a cowpat in (c), negative plants in (d) and (e), tree leaves in (c), and a human foot in (d). Note the color encoding of the two classes: yellow for *Rumex obtusifolius L.* and orange for *Rumex Crispus L.*. The figure is taken from paper C and extended with the additional joint stem annotations from paper D.

The dataset *RumexLeaves* published within paper E is related to the second stage camera which provides a close view RGB image as well as a corresponding depth map of the targeted weed. It captures more fine-grained plant characteristics, such as individual leaf shapes and their corresponding stem and primary vein. Since our robot platform introduced in section 4.1 was not equipped with an accurate localization system, the robot was not able to position itself on top of the plant autonomously. Instead, we collected the data manually by driving the robot over the plant with a remote control. This is time-consuming, therefore, the collected *RoboRumex* datapoints are enriched with *iNaturalist* datapoints, which are images containing *Rumex* plants scraped from the plant publisher iNaturalist[3]. In total, the dataset contains 809 images, where each leaf instance is manually annotated with a segmentation mask for the leaf blade as well as a keypoint-guided polyline leading from the starting point of the stem along the basal to the apex of the leaf. In Figure 4.5, we show sample images and corresponding annotations for the *RoboRumex* and *iNaturalist* datapoints.

[3] https://www.inaturalist.org/

(a) Example iNaturalist datapoints from the *RumexLeaves* dataset



(b) Example RoboRumex datapoints from the *RumexLeaves* dataset

Figure 4.5: Example RGB images (top), depth maps if available (middle), and overlayed annotations (bottom) from the *RumexLeaves* dataset. Pixel-wise leaf blade annotations are shown in orange and polyline annotations of vein and stem are shown in blue. The figure is adapted from paper E.

4.3 Rumex weed detection

The detection of a *Rumex* weed within the field is essential because it is the trigger event for a whole weeding procedure: The detected weed in pixel space is mapped to the 3D space, the robot navigates toward the targeted weed position so that it lies within the working area of the weeding unit. Once the robot stops above the weed, an image is taken with the second stage camera, and relevant traits are extracted from the targeted plant to enable an efficient weeding procedure as described in section 4.4.

For the development of the *Rumex* detection model, the number of incorrect detections, i.e. false positives, plays an important role. Removing false positives, which can be potentially good and nutritive plants, wastes resources and prolongs the working

time of the robot on the field. Moreover, a few missing detections, i.e. false negatives, do not harm the application's success as long as the majority of weeds have been removed successfully. It can be assumed, that the missed weeds will be removed in another session when *Rumex* features have developed further, i.e. the plant grew bigger. Another requirement for the detection model is real-time inference performance on the onboard computing unit, which is the Jetson Xavier NX.

In paper C, we provide baseline detection results with YOLOX [Ge+21], which was the best-performing YOLO architecture at that point in time. We carried out ablation studies concerning the model size, the confidence score, and the influence of having multiple image views of plants in the training dataset. We also showed the generalizability capabilities of the model by training it on sets of field images and evaluating it on images from a different field or a different day.

During the deployment on the robot, we make use of the available video stream and increase the robustness of the plant detection by utilising commonly used tracking algorithms such as the Hungarian Algorithm [Kuh55] based on the Kalman Filter [WB95].

4.4 Fine-grained rumex weed analysis

As described in paper F, once the GALIRUMI robot has positioned itself toward the targeted *Rumex* weed, it retrieves an approximate plant center which is represented by the center coordinates of the enclosing bounding box. This approximation is not very accurate and leads to decreased weeding capabilities for both techniques. Electric weeding is most effective when applied to the joint stem position. Laser weeding is performed by lasering a circle around the bounding box center to showcase the laser weeding capabilities such that it can successfully cut leaves and stems. A more accurate plant center, such as the joint stem position, would decrease the number of missed plant leaves (see the top row of Figure 4.6). Moreover, in practice, the circular laser pattern is not very effective because the laser procedure is fairly slow. After all, lasering a circle of a radius $r = 50mm$ takes approximately 1 minute at a power consumption of up to 200 W. From an economic point of view, this is infeasible because the robot operates too slowly and consumes too much energy per weed removal.

Therefore, in paper D, we address the joint stem detection which can be relevant for both weeding techniques. Multi-task models are developed to detect simultaneously bounding boxes for the *Rumex* plants and keypoints for the corresponding joint stems. It leads to a rough estimate of the weed's size and retrieves a more precise center point of the plant. In paper D, we compare two architectures, both based on the popular YOLOv5 architecture [Gle22], namely YOLO-HRNet and YOLO-Pose. The baseline architecture – YOLO-HRNet – is inspired by MaskRCNN [He+17] and consists of two stages. The input image is processed by the object detection architecture YOLOv5 in the first stage. In the second stage, the feature maps of each detected object are further processed by the HRNet [Wan+21a]. HRNet is specifically designed for key-

point detection and outputs a heatmap, whereas the peak in the heatmap serves as keypoint for the corresponding object. On the contrary, YOLO-Pose [Maj+22] is a regression-based single-stage approach. YOLO-Pose simply extends the YOLOv5 with an additional head for the keypoint prediction, which regresses a keypoint position with respect to the anchor center of the object. Both models are evaluated on the *RumexWeeds* dataset where each *Rumex* weed is manually annotated with a bounding box for the weed enclosing and a keypoint for the corresponding joint stem position, as shown in Figure 4.4. Our results show, that the tasks complement each other because the addition of the joint stem prediction task improves the object detection performance for both architectures. Furthermore, YOLO-Pose outperforms YOLO-HRNet in prediction performance as well as inference speed. We propose to deploy our final YOLO-Pose model to the GALIRUMI robot to provide a better estimate of the joint stem position, thus, improving the performance of both weeding implements. Unfortunately, the scope of the GALIRUMI project didn't allow a deployment, and therefore, an evaluation in the real world remains for future work.

The work from paper D gives a targeted estimate for the joint stem position compared to only considering the center of the predicted bounding box detection. A more exact plant center position allows for a circular laser pattern with a reduced radius, thus, reducing overall power consumption. However, for laser weeding of plants with low leaf density, removing each leaf individually at the lowest visible point could save even more time and energy.

In paper E, we perform a more fine-grained analysis of the *Rumex* weeds, where individual leaves are identified as well as their corresponding stem and vein are extracted. We apply CenterNet [ZWK19], adapted and fine-tuned to our application. Object instances, i.e. leaves, are retrieved from heatmaps, where each peak represents an identified object. Additional characteristics such as leaf bounding box dimensions and stem/vein keypoints are regressed with respect to the object's center point.

The model has been developed with the dataset *RumexLeaves*, which was introduced in section 4.2. We perform ablation studies concerning the best target representation and the network complexity with a special focus on deformable convolutional layers [Zhu+18], which are important for the regression of parameters that deviate a lot from the object's center point. Note, that the stem and vein of the leaves are represented with a keypoint-guided polyline, which has true keypoints with distinct positions at the basal and apex of the leaf blade as well as at the lowest point of the stem. Moreover, it has pseudo-keypoints, which are somewhat evenly spread between the true keypoints on the stem/vein line by the human annotator, however, their position is not distinct and can vary along the line. Hence, a prediction on the line that slightly deviates from the pseudo keypoint should be evaluated as correct in our metrics. To comply with keypoint-guided polylines, we introduce a new metric *Projected OKS* (POKS) in paper E, which is based on the popular OKS metric. POKS projects the predicted pseudo keypoints to the stem/vein line and uses those projections as temporary ground truth for the evaluation with OKS.

The on-field extraction of fine-grained phenotypic information is highly relevant

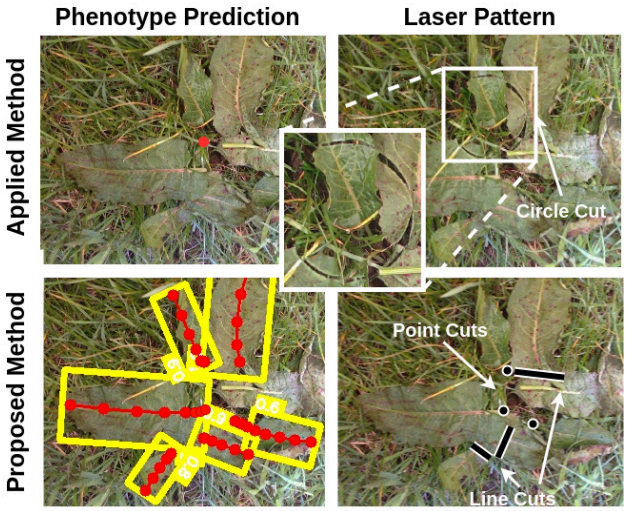


Figure 4.6: Top row: The GALIRUMI application approximates the plant center with the center coordinates of the bounding box of the plant, which is not very accurate due to potential asymmetries of the plant. It leads to an ineffective circle laser pattern, missing a relevant portion of leaves. Bottom Row: We perform a fine-grained plant analysis, where the stem—if visible—and vein of each leaf are extracted. From the stem/vein information (red) as well as the approximate leaf size (yellow), an efficient weeding pattern could be extracted. Please note, that within this work we have not transformed the phenotype predictions to concrete laser patterns, but instead, we added the bottom right figure only to illustrate our idea.

for the domain of field monitoring and breeding. Specifically for our GALIRUMI application, we envision our fine-grained plant analysis to enable more advanced laser patterns as illustrated in Figure 4.6: If the stem is visible a pointcut can be performed at the lowest point, if not, a line cut can be performed at the lowest point of the vein perpendicular to the vein and of the length of the leaf blade width. Moreover, the speed of the laser can be dynamically adjusted, depending on which part of the plant is cut. According to paper F, leaf blades and veins can be cut with much higher velocity compared to stems. Again, the scope of the GALIRUMI project didn't allow further real-world experiments on advanced laser patterns; thus, it remains for future work or subsequent projects.

CHAPTER 5

Conclusion & Future Outlook

In this Ph.D. project a visual perception system was developed to extract plant information on different granularity levels, from coarse plant detections to fine-grained plant traits. The developments of the visual perception system were mainly driven by the EU Horizon 2020 project GALIRUMI supporting this work financially. GALIRUMI's goal was to develop an agricultural robotics platform to control *Rumex obtusifolius/crispus L.* in grassland fields while implementing two new weeding techniques, namely laser- and electric weeding. The project aims to advance digitalization in organic farming, contributing to our present societal challenges, while several UN sustainable development goals are addressed, such as "2 Zero Hunger", "6 Clean Water and Sanitation", "13 Climate Action", "15 Life on Land".

Figure 5.1: The GALIRUMI project aims to advance digitalization in organic farming and addresses several UN sustainable development goals. Icons are taken from https://www.un.org/sustainabledevelopment/.

5.1 Conclusion

While a commercial robot platform was selected as the base platform, the weeding implements needed to be designed and realized from scratch. For the visual perception system, suitable sensors were selected and mounted at appropriate positions to fulfill the field coverage requirements and allow a reasonable interplay between the cameras and weeding units.

Robust deep vision models require large amounts of training data. The publicly available and annotated data for our application was very limited at that point in time. To develop accurate detection models, a reasonable amount of data needed to be collected to properly evaluate the performance. Unfortunately, in the first year of the Ph.D. project, the robot platform was not yet available. Therefore, a limited amount of images was collected with a consumer camera to get the research project started. The images were used to investigate approaches to support the future data collection process with attempts to improve label efficiency in paper B and facilitate the annotation process with the work in paper A, as well as with relevant object tracking algorithms based on the interplay of odometry measurements and visual features.

In the 2nd year, we finalized the robot setup at DTU, which is a slimmed-down setup from the original one situated at Wageningen Research in The Netherlands. The robot contains only a replicate of the visual perception system, but no weeding units or high-precision localization system. We collected the actual data throughout the relevant seasons of the year, i.e. spring, summer, and autumn.

Finally, in the last third of the 2nd year as well as in the 3rd year, we processed a large amount of collected data and used it for vision model development. We created and published two datasets *RumexWeeds* in paper C and *RumexLeaves* in paper E. Both datasets allowed for developing models on different granularity levels: overall *Rumex* weed detection with bounding boxes in paper C, joint stem detection with keypoints in paper D, as well as the extraction of fine-grained plant traits, such as leaf-, stem- and vein instances in paper E.

We have only integrated the coarse detection model on the actual GALIRUMI robot platform. Unfortunately, the time constraints of the project didn't allow for the integration of the more granular perception models with the weeding units. The integration as well as the evaluation in real-world fields remain for future work.

5.2 Future Outlook

In paper F, we present a working prototype robot platform that can navigate on an unstructured field, detect relevant weeds, and treat them with laser techniques. *Rumex* weed detection worked well overall, but on unknown fields, we observed that other broad-leaved weeds such as dandelion (*Taraxacum officinale* L.) and broad-leaf

plantain (*Plantago major* L.) get detected as *Rumex*. We propose a short-term and long-term approach to address the lack of dataset quality.

In the short term, it is infeasible to provide a dataset that is diverse and complete enough to cover all potential field varieties between seasons, years, and locations due to different external influences by weather and light. We have to increase the focus on methods that adapt to new fields at low cost and improve the *RumexWeeds* dataset in the meantime. As a pre-deployment step, the robot could be used to collect a relatively small and representative amount of images for a specific field. The freshly collected data is then used to adapt the base model to that specific field. Hereby, different adapting methods can be considered such as domain adaptation, semi-supervised fine-tuning, and few-shot learning, potentially combined with representation learning. All of the previously mentioned methods require only a small amount of annotated images, which must be provided by the field expert.

In the long run, we need to create a qualitative and emerging dataset to train better base models and thus, reduce the cost during field adaptation. As it is done today for *RumexWeeds* and many related datasets, a fixed number of images, collected on a few different fields throughout one or two seasons, constitutes a static dataset for model training. However, a quality control pipeline should be considered to filter out redundant datapoints, aiming for a careful selection of diverse datapoints at small size. Keeping the dataset size as low as possible has a relevant impact on model training times, leading to lower costs and CO2 impact. Furthermore, strategies must be established to continuously update the *RumexWeeds* datasets with new and enriching datapoints to converge toward completeness over time. Field adaptation as described above extracts candidate datapoints that can be considered to be added to the dataset. Accordingly, models will be updated.

Robot perception is far from what humans are capable of. Humans not only have a complex semantic scene understanding but also extract fine-grained intra-object properties for objects they have never seen before. When humans look at plants, they naturally perceive the plant architecture with its individual leaves and branching system. Our work on fine-grained plant analysis points toward that direction and we see major potential to apply it in other domains apart from complex weeding and phenotyping.

We envision a similar strategy to differentiate between plant species within a field as humans do. The foundation forms a core model that learns the concept of a plant. Regardless of the species, the model extracts plant instances and their traits, such as leaf and flowering -shapes, -orientations, and -coloring as well as the stem and vein branching system with the appropriate thickness and coloring. The core model is trained on a relatively large dataset of different plant species and associated traits, while exact species information is irrelevant to the model. In the second phase, the core model can be used to specialize on any relevant plant species by relating the extracted plant traits to the targeted species. We believe, that this strategy can improve generalizability and is more label-efficient during the field adaptation phase.

Recent research on mixed cropping is combating the established practices of mono-

cultures. Different types of crops and even small portions of weeds that complement each other are grown together in the same field. However, the practice of mixed cropping requires careful and expensive field management, maintaining the right proportion of species within the field. Species that start dominating the composition will need to be controlled. Robots with fine-grained perception capabilities can potentially optimize the control process by not only primitively cutting back a certain proportion of a plant species, but by removing the ones with undesired plant traits.

We believe that an online, robust, and granular plant understanding has the potential to unlock alternative ways for robots to perceive real-world fields. We think that it can enable better generalizability over a large selection of plant species and contribute to the success of scalable mixed crop fields to combat the severe problem of monocultures.

Bibliography

[AA20] Mahmoud Abdulsalam and Nabil Aouf. "Deep Weed Detector/Classifier Network for Precision Agriculture." In: *2020 28th Mediterranean Conference on Control and Automation (MED)*. 2020, pages 1087–1092. DOI: 10.1109/MED48518.2020.9183325.

[AB20] Muhammad Hamza Asad and Abdul Bais. "Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network." In: *Information Processing in Agriculture* 7.4 (2020), pages 535–545. ISSN: 2214-3173. DOI: https://doi.org/10.1016/j.inpa.2019.12.002.

[Ahm+14] Faisal Ahmed et al. "Automated weed classification with local pattern-based texture descriptors." In: *Int. Arab J. Inf. Technol.* 11 (2014), pages 87–94.

[Al-+22] Ahmed Husham Al-Badri et al. "Hybrid CNN Model for Classification of Rumex Obtusifolius in Grassland." In: *IEEE Access* 10 (2022), pages 90940–90957. DOI: 10.1109/ACCESS.2022.3200603.

[Ank+10] Thomas Anken et al. "Automatic detection of broad-leaved dock in grassland." In: January 2010.

[BCF18] A. Binch, N. Cooke, and Charles W Fox. "Rumex and Urtica detection in grassland by UAV." In: *14th International Conference on Precision Agriculture*. 2018.

[Bec+21] Michael Beck et al. *Presenting an extensive lab- and field-image dataset of crops and weeds for computer vision tasks in agriculture*. August 2021.

[Bir+20] Simon Birrell et al. "A field-tested robotic harvesting system for iceberg lettuce." In: *Journal of Field Robotics* 37.2 (2020), pages 225–245. DOI: https://doi-org.proxy.findit.cvt.dk/10.1002/rob.21888.

[Ble+16] Timo Blender et al. "Managing a Mobile Agricultural Robot Swarm for a seeding task." In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pages 6879–6886. DOI: 10.1109/IECON.2016.7793638.

[Bol+20] Daniel Bolya et al. "TIDE: A General Toolbox for Identifying Object Detection Errors." In: *ECCV*. 2020.

[Bos+20] Petra Bosilj et al. "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture." In: *Journal of Field Robotics* 37.1 (2020), pages 7–19. DOI: https://doi.org/10.1002/rob.21869.

[Bot10] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent." In: *Proc. of COMPSTAT* (January 2010). DOI: 10.1007/978-3-7908-2604-3_16.

[BU17] Suchet Bargoti and James Underwood. "Deep fruit detection in orchards." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pages 3626–3633.

[Büh+15] Jonas Bühler et al. "phenoVein—A Tool for Leaf Vein Segmentation and Analysis ." In: *Plant Physiology* 169.4 (October 2015), pages 2359–2370. ISSN: 0032-0889. DOI: 10.1104/pp.15.00974.

[Bus+20] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations." In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125.

[BWL20] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." In: *CoRR* abs/2004.10934 (2020).

[BWS19] Asher Bender, Brett Whelan, and Salah Sukkarieh. "A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica." In: *Journal of Field Robotics* 37 (July 2019). DOI: 10.1002/rob.21877.

[BWS20] Asher Bender, Brett Whelan, and Salah Sukkarieh. "A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica." In: *Journal of Field Robotics* 37.1 (2020), pages 73–96.

[Car+20] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments." In: *Advances in Neural Information Processing Systems*. Edited by H. Larochelle et al. Volume 33. Curran Associates, Inc., 2020, pages 9912–9924.

[Car+23] Angelo Cardellicchio et al. "Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors." In: *Computers and Electronics in Agriculture* 207 (2023), page 107757. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2023.107757.

[CH21] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pages 15750–15758.

[Che+17] Nived Chebrolu et al. "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields." In: *The International Journal of Robotics Research* (2017).

[Che+20] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations." In: *Proceedings of the 37th International Conference on Machine Learning*. Edited by Hal Daumé III and Aarti Singh. Volume 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pages 1597–1607.

[Chi+20] Mang Tik Chiu et al. "Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[CL13] Gabriela Csurka and Diane Larlus. "What is a good evaluation measure for semantic segmentation?" In: *IEEE Trans. Pattern Anal. Mach. Intell.* Volume 26. January 2013. DOI: 10.5244/C.27.32.

[Cor+16] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding." In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Cou+22] Mathias Cougnon et al. "Using perennial plant varieties for use as living mulch for winter cereals. A review." In: *Agronomy for Sustainable Development* 42 (November 2022). DOI: 10.1007/s13593-022-00844-x.

[Cub+20] Sergio Cubero et al. "RobHortic: A Field Robot to Detect Pests and Diseases in Horticultural Crops by Proximal Sensing." In: *Agriculture* 10.7 (2020). ISSN: 2077-0472. DOI: 10.3390/agriculture10070276.

[Cut13] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport." In: *Advances in Neural Information Processing Systems*. Edited by C. J. C. Burges et al. Volume 26. Curran Associates, Inc., 2013.

[Dav+21] Etienne David et al. "Global Wheat Head Detection 2021: An Improved Dataset for Benchmarking Wheat Head Detection Methods." In: *Plant Phenomics* 2021 (2021). DOI: 10.34133/2021/9846158.

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pages 248–255.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of Machine Learning Research* 12.Jul (2011), pages 2121–2159.

[Dij+21] Michiel van Dijk et al. "A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050." In: *Nature Food* 2 (July 2021), pages 494–501. DOI: 10.1038/s43016-021-00322-9.

[Dua+19] Kaiwen Duan et al. "CenterNet: Keypoint Triplets for Object Detection." In: October 2019, pages 6568–6577. DOI: 10.1109/ICCV.2019.00667.

[Dür+08] Lorenz Dürr et al. “Machine vision detection and microwave-based elimination of rumex obtusifolius L. on grassland.” In: 2008.

[Esp+20a] Borja Espejo-Garcia et al. “Improving weeds identification with a repository of agricultural pre-trained deep neural networks.” In: *Computers and Electronics in Agriculture* 175 (2020), page 105593. ISSN: 0168-1699.

[Esp+20b] Borja Espejo-Garcia et al. “Towards weeds identification assistance through transfer learning.” In: *Computers and Electronics in Agriculture* 171 (2020), page 105306. ISSN: 0168-1699.

[Esp+23] Borja Espejo-Garcia et al. “Top-tuning on transformers and data augmentation transferring for boosting the performance of weed identification.” In: *Computers and Electronics in Agriculture* 211 (2023), page 108055. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2023.108055.

[Eve+09] F K van Evert et al. “Real-time vision-based detection of Rumex obtusifolius in grassland.” In: *Weed Research* 49.2 (2009), pages 164–174.

[Eve+11] Frits K. van Evert et al. “A robot to detect and control broad-leaved dock (Rumex obtusifolius L.) in grassland.” In: *Journal of Field Robotics* 28.2 (2011), pages 264–277.

[Eve+20] F K van Evert et al. “Weekly defoliation controls, but does not kill broad-leaved dock (Rumex obtusifolius).” In: *Weed Research* 60.2 (2020), pages 161–170. DOI: https://doi.org/10.1111/wre.12407.

[Faw+21] Mulham Fawakherji et al. “Multi-Spectral Image Synthesis for Crop/Weed Segmentation in Precision Farming.” In: *Robotics and Autonomous Systems* 146 (2021), page 103861. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103861.

[Gar+22] Antoine Gardarin et al. “Intercropping with service crops provides multiple services in temperate arable systems: a review.” In: *Agronomy for Sustainable Development* 42 (June 2022). DOI: 10.1007/s13593-022-00771-x.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[Ge+21] Zheng Ge et al. “YOLOX: Exceeding YOLO Series in 2021.” In: *arXiv preprint arXiv:2107.08430* (2021).

[GEN23] Ronja Güldenring, Frits K. van Evert, and Lazaros Nalpantidis. “RumexWeeds: A grassland dataset for agricultural robotics.” In: *Journal of Field Robotics* (2023). DOI: https://doi.org/10.1002/rob.22196.

[GH10] Michael Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Edited by Yee Whye Teh and Mike Titterington. Volume 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pages 297–304.

[Gle22] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. November 2022. DOI: 10.5281/zenodo.7347926.

[Gri+20] Jean-Bastien Grill et al. "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning." In: *Advances in Neural Information Processing Systems*. Edited by H. Larochelle et al. Volume 33. Curran Associates, Inc., 2020, pages 21271–21284.

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations." In: *International Conference on Learning Representations*. 2018.

[Gül+21] Ronja Güldenring et al. "Few-leaf Learning: Weed Segmentation in Grasslands." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[He+16a] Kaiming He et al. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 770–778. DOI: 10.1109/CVPR.2016.90.

[He+16b] Kaiming He et al. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 770–778.

[He+17] Kaiming He et al. "Mask R-CNN." In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pages 2980–2988. DOI: 10.1109/ICCV.2017.322.

[He+19] Tong He et al. "Bag of Tricks for Image Classification with Convolutional Neural Networks." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pages 558–567.

[He+20] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pages 9726–9735.

[Hin+23] Julia Hindel et al. "INoD: Injected Noise Discriminator for Self-Supervised Representation Learning in Agricultural Fields." In: *IEEE Robotics and Automation Letters* PP (September 2023), pages 1–8. DOI: 10.1109/LRA.2023.3301269.

[HO15] Sebastian Haug and Jörn Ostermann. "A Crop/Weed Field Image Dataset for the Evaluation of Computer Vision Based Precision Agriculture Tasks." In: *Computer Vision - ECCV 2014 Workshops*. 2015, pages 105–116. DOI: 10.1007/978-3-319-16220-1_8.

[Hon86] Akio Hongo. "Infestation of Rumex obtusifolius L. and Distribution Pattern of Its Individual Plants in Sown Grasslands in Eastern Hokkaido." In: *Journal of Weed Science and Technology* 31.4 (1986), pages 300–305.

[Hus+23] Ahmed Husham Al-Badri et al. "Adaptive Non-Maximum Suppression for improving performance of Rumex detection." In: *Expert Systems with Applications* 219 (2023), page 119634. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.119634.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pages 448–456.

[Jac12] Jaccard. "The distribution of the flora of the alpine zone." In: *New Phytologist*. Volume 11. Carg se, France., 1912, pages 37–50.

[JAL23] Kan Jiang, Usman Afzaal, and Joonwhoan Lee. "Transformer-Based Weed Segmentation for Grass Management." In: *Sensors* 23.1 (2023). ISSN: 1424-8220. DOI: 10.3390/s23010065.

[JCC21] Xiaojun Jin, Jun Che, and Yong Chen. "Weed Identification Using Deep Learning and Image Processing in Vegetable Plantation." In: *IEEE Access* 9 (2021), pages 10940–10950. DOI: 10.1109/ACCESS.2021.3050296.

[Jin+22] Xiaojun Jin et al. "A novel deep learning-based method for detection of weeds in vegetables." In: *Pest Management Science* 78.5 (2022), pages 1861–1869. DOI: https://doi.org/10.1002/ps.6804.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *CoRR* abs/1412.6980 (2015).

[Kie+23] Jana Kierdorf et al. "GrowliFlower: An image time-series dataset for GROWth analysis of cauLIFLOWER." In: *Journal of Field Robotics* 40.2 (2023), pages 173–192. DOI: https://doi.org/10.1002/rob.22122.

[KK23] Blaire Kleiman and Suzanne Koptur. "Weeds Enhance Insect Diversity and Abundance and May Improve Soil Conditions in Mango Cultivation of South Florida." In: *Insects* 14.1 (2023). ISSN: 2075-4450. DOI: 10.3390/insects14010065.

[Kon+21] Tukhar Jyoti Konch et al. "Remarkable Rate of Water Evaporation through Naked Veins of Natural Tree Leaves." In: *ACS Omega* 6.31 (2021), pages 20379–20387. DOI: 10.1021/acsomega.1c02398.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems 25*. Edited by F. Pereira et al. Curran Associates, Inc., 2012, pages 1097–1105.

[KTN16] T. Kounalakis, G. Triantafyllidis, and L. Nalpantidis. "Weed recognition framework for robotic precision farming." In: *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*. 2016, pages 466–471.

[KTN18] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Image-Based Recognition Framework for Robotic Weed Control Systems." In: *Multimedia Tools Appl.* 77.8 (April 2018), pages 9567–9594. ISSN: 1380-7501.

[KTN19] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Deep learning-based visual recognition of rumex for robotic precision farming." In: *Computers and Electronics in Agriculture* 165 (2019), page 104973. ISSN: 0168-1699.

[Kuh55] Harold W. Kuhn. "The Hungarian Method for the Assignment Problem." In: *Naval Research Logistics Quarterly* 2.1–2 (March 1955), pages 83–97. DOI: 10.1002/nav.3800020109.

[Lam+17] Petre Lameski et al. "Weed Detection Dataset with RGB Images Taken Under Variable Light Conditions." In: September 2017, pages 112–119. ISBN: 978-3-319-67596-1. DOI: 10.1007/978-3-319-67597-8_11.

[Lam+21] Olee Hoi Ying Lam et al. "An open source workflow for weed mapping in native grassland using unmanned aerial vehicle: using Rumex obtusifolius as a case study." In: *European Journal of Remote Sensing* 54.sup1 (2021), pages 71–88.

[Lan+18] Ferdinand Langer et al. "Geometrical Stem Detection from Image Data for Precision Agriculture." In: *CoRR* abs/1812.05415 (2018).

[LH13] Kuebum Lee and K.-S Hong. "An implementation of leaf recognition system using leaf vein and shape." In: *International Journal of Bio-Science and Bio-Technology* 5 (January 2013), pages 57–65.

[LH17] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam." In: *CoRR* abs/1711.05101 (2017).

[Li+11] C. Li et al. "Extraction of leaf vein based on improved Sobel algorithm and hue information." In: *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering* 27 (July 2011), pages 196–199. DOI: 10.3969/j.issn.1002-6819.2011.07.034.

[Lin+14] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context." In: *CoRR* abs/1405.0312 (2014).

[Lin+17] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection." In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pages 2999–3007. DOI: 10.1109/ICCV.2017.324.

[Liu+21] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[Liu+22] Xiaobao Liu et al. "Plant leaf veins coupling feature representation and measurement method based on DeepLabV3+." In: *Frontiers in Plant Science* 13 (2022). ISSN: 1664-462X. DOI: 10.3389/fpls.2022.1043884.

[Lot+18] Philipp Lottes et al. "Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming." In: June 2018. DOI: 10.1109/IROS.2018.8593678.

[Lot+20] Philipp Lottes et al. "Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming." In: *Journal of Field Robotics* 37.1 (January 2020), pages 20–34. ISSN: 1556-4959. DOI: 10.1002/rob.21901.

[Luk+18] Alan Lukežič et al. "Discriminative Correlation Filter with Channel and Spatial Reliability." In: *International Journal of Computer Vision* 126 (July 2018). DOI: 10.1007/s11263-017-1061-3.

[Maj+22] Debapriya Maji et al. *YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss*. 2022. DOI: 10.48550/ARXIV.2204.06806.

[Mar+20] Paula A Marin Zapata et al. "Self-supervised feature extraction from image time series in plant phenotyping using triplet networks." In: *Bioinformatics (Oxford, England)* (October 2020). ISSN: 1367-4803.

[MMS22] Elias Marks, Federico Magistri, and Cyrill Stachniss. "Precise 3D Reconstruction of Plants from UAV Imagery Combining Bundle Adjustment and Template Matching." In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pages 2259–2265. DOI: 10.1109/ICRA46639.2022.9811358.

[MYA17] Mostafa Mehdipour Ghazi, Berrin Yanikoglu, and Erchan Aptoula. "Plant identification using deep neural networks via optimization of transfer learning parameters." In: *Neurocomputing* 235 (2017), pages 228–235. ISSN: 0925-2312.

[OES23] Franklin C. Ogidi, Mark G. Eramian, and Ian Stavness. "Benchmarking Self-Supervised Contrastive Learning Methods for Image-Based Plant Phenotyping." In: *Plant Phenomics* 5 (2023), page 0037. DOI: 10.34133/plantphenomics.0037.

[Ola+21] Olayemi Mikail Olaniyi et al. "Weed Recognition System for Low-Land Rice Precision Farming Using Deep Learning Approach." In: *Artificial Intelligence and Industrial Applications*. Edited by Tawfik Masrour, Anass Cherrafi, and Ibtissam El Hassani. Cham: Springer International Publishing, 2021, pages 385–402. ISBN: 978-3-030-51186-9.

[Ols+19] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning." In: *Scientific Reports* 9.2058 (1 February 2019). DOI: 10.1038/s41598-018-38343-3.

[OLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding." In: *ArXiv* abs/1807.03748 (2018).

[Pan+22] Ling Pan et al. "Coordination of stomata and vein patterns with leaf width underpins water-use efficiency in a C4 crop." In: *Plant, Cell & Environment* 45.6 (2022), pages 1612–1630. DOI: https://doi.org/10.1111/pce.14225.

[Par+] Soran Parsa et al. "Modular autonomous strawberry picking robotic system." In: *Journal of Field Robotics* n/a.n/a (). DOI: https://doi-org.proxy.findit.cvt.dk/10.1002/rob.22229.

[Pir+19] Taihú Pire et al. "The Rosario dataset: Multisensor data for localization and mapping in agricultural environments." In: *The International Journal of Robotics Research* 38.6 (2019), pages 633–641.

[Pol+07] G. Polder et al. "Weed Detection Using Textural Image Analysis." English. In: *EFITA/WCCA Conference*. 2007.

[RF18] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. cite arxiv:1804.02767Comment: Tech Report. 2018.

[Rog+23a] Gianmarco Roggiolani et al. "Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain." In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. 2023.

[Rog+23b] Gianmarco Roggiolani et al. "On Domain-Specific Pre- Training for Effective Semantic Perception in Agricultural Robotics." In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pages 11786–11793. DOI: 10.1109/ICRA48891.2023.10160624.

[Rom+18] Eduardo Romera et al. "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation." In: *IEEE Transactions on Intelligent Transportation Systems* 19.1 (2018), pages 263–272. DOI: 10.1109/TITS.2017.2750080.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Volume 9351. LNCS. Springer, 2015, pages 234–241.

[S H04] P. Doležal S. Hejduk. "Nutritive value of broad-leaved dock (Rumex obtusifolius L.) and its effect on the quality of grass silages." In: *Czech Journal of Animal Science* 49 (2004), pages 144–150.

[Sa+18] Inkyu Sa et al. "WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming." In: *Remote Sensing* 10.9 (2018). ISSN: 2072-4292. DOI: 10.3390/rs10091423.

[SAŠ19] Damian Schori, Thomas Anken, and Dejan Šeatović. "Using Fully Convolutional Networks for Rumex Obtusifolius Segmentation, a Preliminary Report." In: *2019 International Symposium ELMAR*. 2019, pages 119–122.

[SCB21] Guruprasad Samanta, Amlan Chakrabarti, and Bhargab Bhattacharya. "Extraction of Leaf-Vein Parameters and Classification of Plants Using Machine Learning." In: January 2021, pages 579–586. ISBN: 978-981-15-7833-5. DOI: 10.1007/978-981-15-7834-2_54.

[Sek+20] Boris Sekachev et al. *opencv/cvat: v1.1.0*. Version v1.1.0. August 2020. DOI: 10.5281/zenodo.4009388.

[Sel+17] Jesse Dave S. Selda et al. "Plant Identification by Image Processing of Leaf Veins." In: *Proceedings of the International Conference on Imaging, Signal Processing and Communication*. ICISPC 2017. Penang, Malaysia: Association for Computing Machinery, 2017, pages 40–44. ISBN: 9781450352895. DOI: 10.1145/3132300.3132315.

[SIV16] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." In: *CoRR* abs/1602.07261 (2016).

[Sko+19] Søren Skovsen et al. "The GrassClover Image Dataset for Semantic and Hierarchical Species Understanding in Agriculture." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pages 2676–2684.

[Smi+19] Olivia M. Smith et al. "Organic Farming Provides Reliable Environmental Benefits but Increases Variability in Crop Yields: A Global Meta-Analysis." In: *Frontiers in Sustainable Food Systems* 3 (2019). ISSN: 2571-581X. DOI: 10.3389/fsufs.2019.00082.

[Soz+22] Marco Sozzi et al. "Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms." In: *Agronomy* 12.2 (2022). ISSN: 2073-4395. DOI: 10.3390/agronomy12020319.

[Sri+14] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *J. Mach. Learn. Res.* 15.1 (January 2014), pages 1929–1958. ISSN: 1532-4435.

[SSP22] Georg Roman Schneider, Josef Scharinger, and Claudia Probst. "Weed Detection in Grassland and Field Areas Employing RGB Imagery with a Deep Learning Algorithm Using Rumex obtusifolius Plants as a Case Study." In: *Engineering Proceedings* 27.1 (2022). ISSN: 2673-4591. DOI: 10.3390/ecsa-9-13950.

[Ste+23] Daniel Steininger et al. "The CropAndWeed Dataset: A Multi-Modal Learning Approach for Efficient Crop and Weed Manipulation." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. January 2023, pages 3729–3738.

[Suh+18] Hyun K. Suh et al. "Transfer learning for the classification of sugar beet and volunteer potato under field conditions." In: *Biosystems Engineering* 174 (2018), pages 50–65. ISSN: 1537-5110.

[SWS08] Dejan Seatović, Zhaw Winterthur, and Switzerland. "3D-Object Recognition, Localization and Treatment of Rumex Obtusifolius in its Natural Environment." In: *International Conference on Precision Agriculture*. June 2008.

[SZ14] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *CoRR* abs/1409.1556 (2014).

[Sze+15] Christian" "Szegedy et al. "Going deeper with convolutions." In: 2015.

[Tei+18] Nima Teimouri et al. "Weed Growth Stage Estimator Using Deep Convolutional Neural Networks." In: *Sensors* 18.5 (2018). ISSN: 1424-8220. DOI: 10.3390/s18051580.

[Tia+19] Zhi Tian et al. "FCOS: Fully Convolutional One-Stage Object Detection." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pages 9626–9635.

[Val+19] J. Valente et al. "DETECTING RUMEX OBTUSIFOLIUS WEED PLANTS IN GRASSLANDS FROM UAV RGB IMAGERY USING DEEP LEARNING." In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W5 (2019), pages 179–185.

[Val+22] João Valente et al. "Mapping of Rumex obtusifolius in nature conservation areas using very high resolution UAV imagery and deep learning." In: *International Journal of Applied Earth Observation and Geoinformation* 112 (2022), page 102864. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2022.102864.

[Van+21] Grant Van Horn et al. "Benchmarking Representation Learning for Natural World Image Collections." English. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Conference on Computer Vision and Pattern Recognition 2021, CVPR 2021 ; Conference date: 19-06-2021 Through 25-06-2021. United States: Institute of Electrical and Electronics Engineers (IEEE), November 2021,

pages 12884–12893. ISBN: 978-1-6654-4510-8. DOI: 10.1109/CVPR46437.2021.01269.

[Vee+20] Arun Narenthiran Veeranampalayam Sivakumar et al. "Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery." In: *Remote Sensing* 12.13 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12132136.

[Wan+14] Jiang Wang et al. "Learning Fine-Grained Image Similarity with Deep Ranking." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pages 1386–1393.

[Wan+19a] Chien-Yao Wang et al. "CSPNet: A New Backbone that can Enhance Learning Capability of CNN." In: *CoRR* abs/1911.11929 (2019).

[Wan+19b] Kaixin Wang et al. "PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment." In: *The IEEE International Conference on Computer Vision (ICCV)*. October 2019.

[Wan+20] Yuqing Wang et al. "CenterMask: Single Shot Instance Segmentation With Point Representation." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pages 9310–9318. DOI: 10.1109/CVPR42600.2020.00933.

[Wan+21a] J. Wang et al. "Deep High-Resolution Representation Learning for Visual Recognition." In: *IEEE Transactions on Pattern Analysis amp; Machine Intelligence* 43.10 (October 2021), pages 3349–3364. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2020.2983686.

[Wan+21b] Xinlong Wang et al. "Dense Contrastive Learning for Self-Supervised Visual Pre-Training." In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. 2021.

[WB95] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. Technical report 95-041. Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, 1995.

[WD21] Less Wright and Nestor Demeure. "Ranger21: a synergistic deep learning optimizer." In: *CoRR* abs/2106.13731 (2021).

[Wey+21] Jan Weyler et al. "Joint Plant Instance Detection and Leaf Count Estimation for In-Field Plant Phenotyping." In: *IEEE Robotics and Automation Letters* 6.2 (2021), pages 3599–3606. DOI: 10.1109/LRA.2021.3060712.

[Wey+22a] J. Weyler et al. "In-Field Phenotyping Based on Crop Leaf and Plant Instance Segmentation." In: 2022.

[Wey+22b] J. Weyler et al. "Joint Plant and Leaf Instance Segmentation on Field-Scale UAV Imagery." In: 7.2 (2022), pages 3787–3794. ISSN: 377-3766. DOI: 10.1109/LRA.2022.3147462.

[Wey+23] Jan Weyler et al. "PhenoBench — A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain." In: *arXiv preprint* (2023).

[Yan+21] Ceyuan Yang et al. "Instance Localization for Self-supervised Detection Pretraining." In: *CVPR*. 2021.

[YB15] Jiaqian Yu and Matthew Blaschko. "Learning Submodular Losses with the Lovasz Hinge." In: *Proceedings of the 32nd International Conference on Machine Learning*. Edited by Francis Bach and David Blei. Volume 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pages 1623–1631.

[YLL20] Jie You, Wei Liu, and Joonwhoan Lee. "A DNN-based semantic segmentation for detecting weed and crop." In: *Computers and Electronics in Agriculture* 178 (2020), page 105750. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105750.

[Zal04] J G Zaller. "Ecology and non-chemical control of Rumex crispus and R. obtusifolius (Polygonaceae): a review." In: *Weed Research* 44 (2004), pages 414–432.

[Zbo+21] Jure Zbontar et al. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction." In: *Proceedings of the 38th International Conference on Machine Learning*. Edited by Marina Meila and Tong Zhang. Volume 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pages 12310–12320.

[Zha+17] Hengshuang Zhao et al. "Pyramid Scene Parsing Network." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pages 6230–6239. DOI: 10.1109/CVPR.2017.660.

[Zha+18] W. Zhang et al. "Broad-Leaf Weed Detection in Pasture." In: *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. 2018, pages 101–105.

[Zha+20a] Xiaoguang Zhang et al. "A Unified Model for Real-Time Crop Recognition and Stem Localization Exploiting Cross-Task Feature Fusion." In: *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. 2020, pages 327–332. DOI: 10.1109/RCAR49640.2020.9303270.

[Zha+20b] Zhicheng Zhao et al. "When Self-Supervised Learning Meets Scene Classification: Remote Sensing Scene Classification Based on a Multitask Learning Framework." In: *Remote Sensing* 12.20 (2020). ISSN: 2072-4292.

[Zha+21] Yifu Zhang et al. "Fairmot: On the fairness of detection and re-identification in multiple object tracking." In: *International Journal of Computer Vision* 129 (2021), pages 3069–3087.

[Zhu+18] Xizhou Zhu et al. "Deformable ConvNets v2: More Deformable, Better Results." In: *CoRR* abs/1811.11168 (2018).

[Zhu+20] Jiyou Zhu et al. "A Fast and Automatic Method for Leaf Vein Network Extraction and Vein Density Measurement Based on Object-Oriented Classification." In: *Frontiers in Plant Science* 11 (2020). ISSN: 1664-462X. DOI: 10.3389/fpls.2020.00499.

[ZW10] Xiaodong Zheng and Xiaojie Wang. "Leaf Vein Extraction Based on Gray-scale Morphology." In: *International Journal of Image, Graphics and Signal Processing* 2 (December 2010). DOI: 10.5815/ijigsp.2010.02.04.

[ZWK19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as Points." In: *CoRR* abs/1904.07850 (2019).

PAPER A

Few-leaf learning: Weed segmentation in grasslands

Ronja Güldenring[1,*], Evangelos Boukas[1], Ole Ravn[1] and Lazaros Nalpantidis[1]

[*]Corresponding author: ronjag@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmarks

Abstract

Autonomous robotic weeding in grasslands requires robust weed segmentation. Deep learning models can provide solutions to this problem, but they need to be trained on large amounts of images, which in the case of grasslands are notoriously difficult to obtain and manually annotate. In this work we introduce *Few-leaf Learning*, a concept that facilitates the training of accurate weed segmentation models and can lead to easier generation of weed segmentation datasets with minimal human annotation effort. Our approach builds upon the fact that each plant species within the same field has relatively uniform visual characteristics due to similar environmental influences. Thus, we can train a field-and-day-specific weed segmentation model on synthetic training data stemming from just a handful of annotated weed leaves. We demonstrate the efficacy of our approach for different fields and for two common grassland weeds: Rumex obtusifolius (broad-leaved dock) and Cirsium vulgare (spear thistle). Our code is publicly available at https://github.com/RGring/WeedAnnotator.

A.1 Introduction

Precision farming can pave the way towards more sustainable agricultural production. While different robots are being deployed in the fields to handle crops or control weeds, what the vast majority of those robots have in common is that they employ deep learning-based computer vision systems for plant detection. Deep learning constitutes a strong tool for detection but comes at a significant cost: large amounts of annotated training data are required. In crop cultivation, plants are clearly separable from the soil by means of simple color thresholding or by considering near-infrared (NIR) imagery that has higher reflectivity on chlorophyll content. However, this is not an option in grassland farming, as foreground vegetation (weeds) and background (grass) share similar color and chlorophyll content. Manual annotation of grassland data becomes an extremely laborious task and thus, such publicly available datasets are scarce. To the best of our knowledge, there is no real-world grassland dataset with pixel-level annotations publicly available.

This work introduces the concept of *Few-leaf Learning* which can generate field-and-day-specific segmentation models from just a handful of weed leaves. But how can training with just a few leaves, as shown in Fig. A.1, lead to accurate weed segmentation? We build upon the observation that the visual appearance—phenotype—of plants belonging to a *specific species*—same genotype, captured within *one field and one day*, has relatively low variance. This is mainly because plants within the same field have been exposed to the same environmental variations, which restrains the expression of phenotypic plasticity and thus results in similar leaf color, size, shape and thickness. As an example, Fig. A.1 shows nine randomly drawn leaves from

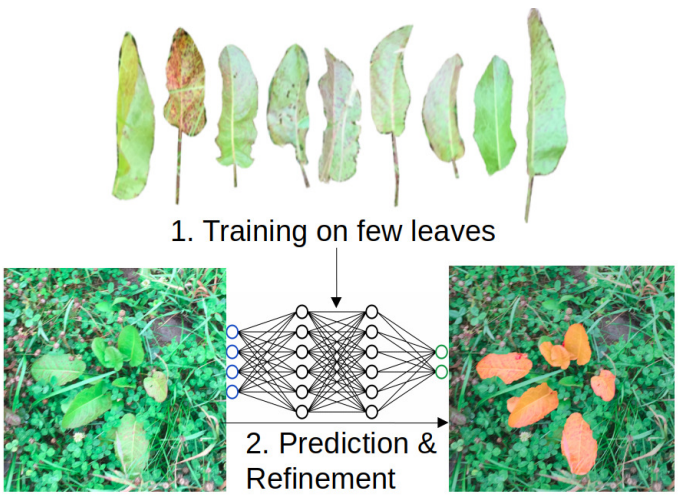


Figure A.1: Few-leaf Learning: A segmentation model can be trained on a very small number of manually annotated leaves.

one of our data collection sessions, providing similar features in shape, texture and color. We train semantic segmentation models on synthetic data generated using standard image composition techniques. Hereby, a very small number of weed leaves needs to be manually annotated and then pasted on weed-free grassland background images, generating numerous combinations of images. The model learns obviously relevant weed features, but also learns to separate those features from the variety of possible background features occurring in other plants, grass and mud. Despite our intended naming similarity, Few-leaf Learning is not related to the well-known *Few-shot Learning*, because in our case the network can and should be trained on a rather large number of synthetic training samples–though, just using only a few leaves to generate them.

Our work can significantly simplify the creation of annotated grassland datasets. We show that each time a data collection session has been performed on a field, Few-leaf Learning can be applied to predict the majority of weed occurrences within that session. Afterwards, the human annotator can additionally perform corrections or further annotations, as needed. We hope that our approach will encourage more publications of high-quality grassland datasets in the future, which is also our own ambition.

The contributions of this work are summarized as follows:

- We experimentally show, that nine manually annotated leaves seem to be sufficient for the proposed Few-leaf Learning.
- We refine the mask predictions from the Few-leaf learning by applying GrabCut in an automatic manner (i.e. no user input is required). This leads to a notable performance improvement of ≈7% for the *Intersection of Union* as well as the *Boundary F1 score*.
- We show the generalizability of our final Few-leaf Learning pipeline by applying it to four different data collection sessions of different locations, days, and plants, i.e. Rumex obtusifolius (broad-leaf dock) and Cirsium vulgare (spread thistle).
- We make our code publicly available at `https://github.com/RGring/WeedAnnotator`.

A.2 Related Work

A.2.1 Agricultural Datasets

The automated detection of weeds in grassland has been investigated in agricultural robotics since the early 2000s, e.g. [Pol+07; Eve+09; Ahm+14; KTN16; KTN18]. However, the proposed approaches are difficult to compare, because the research groups evaluated the approaches on their own unpublished datasets. In total three relevant published grassland/weed datasets were identified:

- *Rumex Ancenis dataset* [KTN19] including monochrome images of Rumex obtusifolius with bounding box annotations. The images were collected during one day and field under controlled lighting conditions.
- *GrassClover dataset* [Sko+19] including synthetic grass-clover images using image composition.
- *DeepWeeds dataset* [Ols+19] including 8 common Australian weeds in their natural environment. The annotations are on a classification level.

None of above provide pixel-level annotations on real-world images. This is a clear gap. In contrast, publicly available crop field datasets are of higher quality and most of them include annotations on a pixel-level which is relevant information for the weeding process, like e.g. cauliflower/broccoli crop dataset [BWS19], sugar beet crop dataset [Che+17] and carrot crop dataset [HO15].

A.2.2 Semi-automatic Annotation

The objective of semi-automatic annotation is to reduce the amount of human input while maintaining the same annotation quality. This can especially save a significant amount of time on pixel-level annotations.

In interactive segmentation, masks are predicted based on negative and positive input provided by the annotator. Ideally, it requires only a few clicks to obtain high-quality segmentation masks. GrabCut [RKB04] is one of the pioneering works, formulated as an optimization problem. It requires bounding box input in order to generate a pixel-level annotation. The initial annotation proposal can be further refined with additional user input in the form of scribble lines. More recently, interactive segmentation using deep learning techniques outperformed previous approaches, such as Deep Extreme Cut [Man+18], f-BRS [Sof+20] and Inside Out Guidance [Zha+20].

Finally, another way of automating the annotation process is to automatically generate object proposals. The model generates them based on unlabeled or weakly-labeled data. Finally, the annotator corrects and complements the given object proposals to conclude the annotation process [Pap+16; RLF15].

A.2.3 Synthetic Image Generation

Synthetic images are artificial images, that have been generated programmatically. The great advantage of using synthetic data is the cost-efficiency of generating large amounts of data. There is no need to collect data in the real world and more importantly, there is no need to annotate the generated data.

Using image composition, new images are generated by pasting foreground objects to new background images. Dwibedi et. al. [DMH17] pastes kitchen objects in new scenes. Their focus is patch-level realism, i.e. handling pasting artifacts. Tripathi et. al. [Tri+19] tackle global realism by extending the basic image composition

with a GAN network that learns where to position the foreground object in the scene. In agriculture, image composition has been successfully applied in the area of seed phenotyping [Tod+20], leaf segmentation and counting [Kuz+19; WMH18] and clover-grass segmentation [Sko+19].

Another way of generating synthetic images is to render samples from a simulation environment, e.g. Cicco et. al. [Di +17] simulate sugar beet leaves as kinematic chains, combining with real-world textures. Building up such realistic simulations is complex and needs expert knowledge. Therefore, the costs might be shifted from manual annotation to creating appropriate simulation environments. Furthermore, the unrealistic appearance of rendered images is still a major issue. Domain Randomization [And+20; Tob+17] as well as Domain adaptation (mainly GANs [Shr+17; Zhu+17; Tsa+18]) are popular approaches to close the simulation-to-reality gap.

A.3 Proposed Pipeline

Our Few-leaf Learning pipeline consists of three main steps:

A Generation of synthetic data by pasting a small number of weed leaves on top of weed-free background images.

B Generation of weed mask predictions using a semantic segmentation model trained on the synthetic data only.

C Mask prediction refinement using GrabCut[RKB04] in an automated manner.

A.3.1 Synthetic image generation

First, synthetic images are generated using image composition methodologies. As input, we provide a small number of precisely segmented leaves L and background field images B that do not contain the targeted plant. Additionally, plant parameters are required like the range of the number of leaves $[p_l, p_u]$ and the maximum distance to the root center d_r. Finally, the synthetic images are composed according to the procedure described in Algorithm 1. We do not consider pure background images ($np > 0$) because the generated foreground images include already a significant amount of background pixels. In Fig. A.2, example synthetic images are presented for both weeds – Rumex obtusifolius and Cirsium vulgare.

Previous studies showed, that local pasting artifacts can reduce the performance of detection methods [DMH17; Tri+19]. Gaussian filters smooth the sharp object boundaries. Furthermore, Poisson image blending [PGB03] keeps the edges from the source patch, but adapts the colors from the target image (mode: *Normal*). An extension is to keep the stronger edges from either the source or the target (mode: *Mix*). We consider two variants of Poisson image blending. In variant *Poisson Normal*, we simply apply Poisson image blending in mode *normal* to the segmented leaf and the background image. In variant *Poisson Mix + Normal*, we first blend the leaf as a

Figure A.2: Examples of synthetic image compositions: top: Rumex obtusifolius, Right bottom: Cirsium vulgare

bounding box in mode *mix* and second add the segmented leaf in mode *normal*. This merges the grass structure of the target batch with the grass structure of the source image while keeping the proper texture of the leaf. Figure A.3 shows the global and local effects of both Poisson blending variants.

In order to avoid blending artifacts being associated with the targeted weed, negative objects are pasted to the image [DMH17; Tri+19]. As a consequence, blending artifacts will be present both in the background and foreground. We consider two types of distractor objects. The *grass distractor* is a randomly selected grass leaf object, that will be added to the image composition. The *shape distractor* [Tri+19] has the shape of one of the input leaves, but contains the texture and color of a random cutout in another randomly selected background image. A positive side-effect of adding distractor objects is the generation of partially occluded foreground objects.

A.3.2 Model Training

We use the PSPNet network architecture [Zha+17] with Resnet18 [He+16] as the backbone to train a semantic segmentation model on the synthetically generated images. We restrict the model size with a Resnet18 in order to avoid overfitting to the small number of different leaves.

Algorithm 1 Image composition procedure

Input: Field background images B
Input: Upwards pointing leaves L
Input: range of number of leaves $[p_l, p_u]$, max root center dist d_r
Output: Composed image

$B_i \Leftarrow$ random_select(B) ▷ *Background image*
$B_i \Leftarrow$ random_flip(random_scale(random_rotate(B_i)))
$n_p \Leftarrow$ random_select([1, 2, 3, 4]) ▷ *Number of plants*

for $i = 0$ to n_p **do**
  $x_i, y_i \Leftarrow$ random_select_position(B_i)
  $n_l \Leftarrow$ random_select($[p_l, ..., p_u]$)) ▷ *Number of leaves*

  for $j = 0$ to n_l **do**
    $L_j \Leftarrow$ random_select(L)
    $d_r \Leftarrow$ random_select($[0, ..., d_r]$)
    $\theta_r \Leftarrow$ random_select($[0, ..., 360]$)
    Paste L_j to B_i with angle θ_r and distance d_r to $[x_i, y_i]$
    Optional: Apply blending method
  end for
  Optional: Add distractor objects to B_i
end for

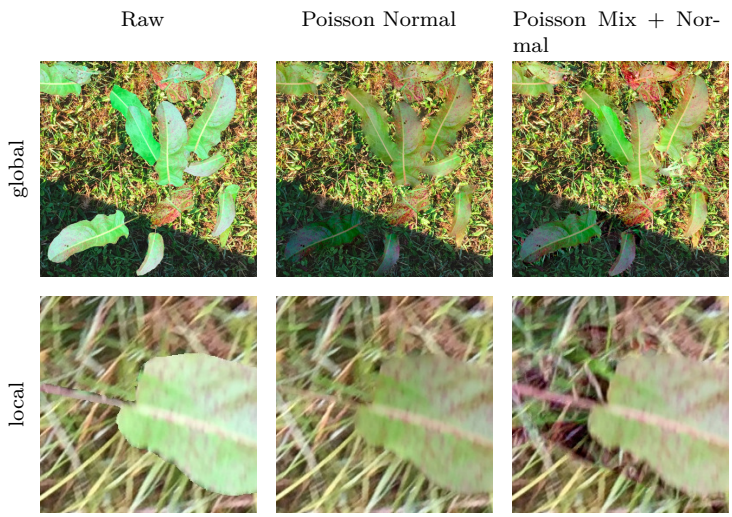


Figure A.3: Effects of Poisson blending [PGB03]. Globally, both Poisson blending variants adapt to the color ambiance of the background image. Locally, variant *Poisson Mix + Normal* merges the grass textures of the source patch and the target image.

A.3.3 Mask refinement

We investigate the usage of two classic interactive segmentation algorithms GrabCut [RKB04] and marker-based watershed [Mey92] to refine the model predictions. Originally, classic interactive segmentation algorithms expect the user to manually provide information about background and foreground regions. Instead, we use the prediction generated by the trained PSPNet to initialize background and foreground areas. This removes the necessity of user inputs and automates the methods. Concretely, our mask refinement consists of two steps as shown in Figure A.4: (1) Mask shrinking by applying morphological erosion to the background and foreground area of the model prediction; (2) Applying classical interactive segmentation like GrabCut [RKB04] or marker-based watershed [Mey92] to the eroded mask and the corresponding image. Finally, we remove noise using morphological closing.

A.4 Experimental Setup

A.4.1 Data Acquisition

For this work, we have collected four sets of data. Each collection has been gathered on one field within one hour so that we can expect a relatively low variance of leaf features, which makes the proposed Few-leaf Learning applicable. The images were acquired using a 8-megapixel 1/3-inch camera with a resolution of 2448×3263 pixels while freely moving through the field. As a consequence, the collected images contain different levels of blur.

In data collection (dc) 1, 2, and 3 the weed *Rumex obtusifolius* has been manually annotated per leaf, i.e. for each leaf an individual instance has been created. In data collection 4, the weed *Cirsium vulgare* has been annotated per plant, as the plants are quite dense and it is most often infeasible for the annotator to identify all single

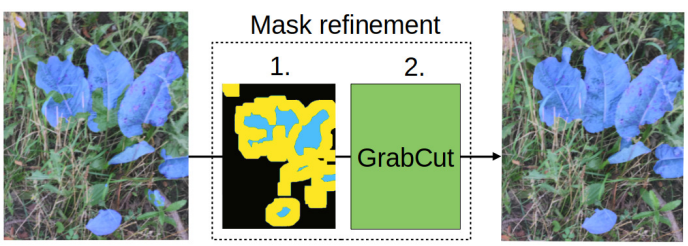


Figure A.4: The mask refinement consists of two steps: 1. The predicted mask will be shrunk by applying morphological erosion to the background (black) and foreground (blue), resulting in a third unknown area (yellow). 2. The eroded mask initializes classical interactive segmentation algorithms like GrabCut[RKB04] or watershed [Mey92] to fill the unknown area.

leaves. Table C.1 gives an overview of the data collections. For each gathering, the number of images containing instances of the considered weed (FG—foreground) and the number of weed-free field images (BG—background) is provided, as well as the average leaf/plant pixel size as a percentage of the whole image pixel size. In the case of dc4, the average object size is significantly bigger, as the annotation was performed per plant, rather than per leaf.

Table A.1: Summary of the four data collections (dc)

ID	dc1	dc2	dc3	dc4
Weed	Rumex obtusifolius			Cirsium vulgare
Date	Sep. 20, 2020	Sep. 03, 2020	Sep. 12, 2020	Sep. 20, 2020
Location	55°44'43.7"N 12°22'53.9"E	55°46'25.6"N 12°32'16.2"E	55°47'20.3"N 11°56'40.5"E	55°44'44.1"N 12°26'10.3"E
# FG imgs (train/val/test)	73 (36/18/19)	109 (54/27/28)	43 (7/18/18)	72 (6/33/33)
# BG imgs (train/val/test)	120 (60/30/30)	120 (60/30/30)	59 (27/16/16)	99 (50/24/25)
Annotation	leaf-based			plant-based
# Objects	589	619	329	125
# leaf pool	75	93	58	25
avg. object size	0.51 %	0.30 %	0.52 %	2.64 %

For the generation of synthetic training images, a pool of complete non-occluded leaves is extracted from the training split (see *# leaf pool* in table C.1). Hereby, blurry leaves are discarded. The synthetic training images are generated by randomly drawing n number of leaves from that pool and composing them with the background images from the training split. Validation and test results are reported on the real data, that has been annotated manually.

A.4.2 Metrics

Three standard metrics are considered to evaluate the semantic segmentation performance. The global accuracy (GAcc) computes the pixel accuracy over the dataset. Its main disadvantage is misleading scores on imbalanced datasets. Also, we consider the widely known Intersection over Union (IoU) only for the foreground objects and call it therefore *Plant IoU*. While the IoU addresses the region similarity, the Boundary F1 score (BF score) [CL13] evaluates the contour similarity per image. Figure A.5 shows example cases of the IoU and BF-score for different mask predictions.

Figure A.5: IoU and BF-score for predictions of different mask qualities. Note, that the framed mask represents the ground truth (IoU=BF=1) with over-segmentations to the left and under-segmentations to the right.

A.4.3 Training configuration

For all experiments, the same training configuration is applied. The Resnet18 backbone of the PSP-Network is initialized with weights, pre-trained on ImageNet [Den+09]. The network is trained on 50 synthetically generated images (downscaled to 612×816 pixels) for 300 epochs. It is updated using the Adam Optimizer [KB15] with a learning rate = 0.0001 and the Lovasz Loss [BB17]. The following image-based augmentations were applied: random flip, random scale, random rotation, random crop, and motion blur. Note, that during validation the last model state (at epoch 300) will be considered. Likewise, during the application of Few-leaf Learning, there will be no ground-truth validation split available to perform early stopping to prevent overfitting.

A.5 Experimental Evaluation

We have conducted four experiments to evaluate the proposed Few-leaf Learning. In the first three, we are considering just the training- and validation- split of data collection 1 to define the best set of hyperparameters. This collection provides leaf-based annotations, allowing in-depth evaluations. Then, in the the fourth experiment, we are using these outcomes and apply our pipeline to the test splits of all four data collections to evaluate the generalization capabilities over different fields, days, and plants.

A.5.1 Blending Methods: Ablation Study

In this experiment, the effect of different blending methods is investigated. We are considering the methods discussed in Sec. A.3.1, as well as a number of combinations of those, where each leaf has been blended with a certain probability:
- *Blending Mix1:* [R: 0.5, GB: 0.5, P_N: 0.0, P_{M+N}: 0.0],
- *Blending Mix2:* [R: 0.33, GB: 0.33, P_N: 0.0, P_{M+N}: 0.34],
- *Blending Mix3:* [R: 0.25, GB: 0.25, P_N: 0.25, P_{M+N}: 0.25].

We have used 9 segmented leaves to generate 50 image compositions with the different considered blending methods and additional distractor objects. Table A.2 shows our considered metrics (GAcc, Plant IoU and BF) obtained for the different

blending methods. Most blending methods do not have a significant influence on the model performance. Considered in isolation, *Poisson Mix + Normal* performs best but mixing it with raw pasting and Gaussian edge smoothing leads to even better performance. Thus, in all further experiments, *Blending Mix2* will be used.

Table A.2: Results on the validation split of data collection 1 for models trained on image compositions with different blending methods.

Blending method	GAcc	Plant IoU	BF
Raw (R)	0.984	0.598	0.651
Gaussian Blur (GB)	0.984	0.585	0.627
Poisson Normal (P_N)	0.979	0.467	0.432
Poisson Mix + Normal (P_{M+N})	0.985	0.617	0.670
Blending Mix1	0.984	0.594	0.646
Blending Mix2	0.985	**0.624**	0.684
Blending Mix3	0.985	0.620	0.657
Blending Mix2 + shape distractor	0.984	0.610	0.666
Blending Mix2 + grass distractor	0.985	0.619	**0.686**

A.5.2 How few leaves are required?

In this experiment, the minimum required number of leaves is determined in order to predict the majority of the foreground objects. From the leaf pool of intact leaves (see table C.1), we draw randomly $n = 1, 3, 9, 15, 21$ leaves to generate synthetic image compositions. For each number of n leaves, we repeat the experiment 10 times. Figure A.6 shows the mean and variance of Plant IoU and BF-score over the drawn number of leaves. The Plant IoU and BF-score both increase with increasing number of leaves and saturate at some point—this appears to be 9 leaves.

A.5.3 Effectiveness of mask refinement

In this experiment, we want to show the impact of the mask refinement step, where we apply GrabCut as well as watershed to the mask predictions generated by PSPNet. A model trained with 9 different leaves with an average performance (Plant IoU = 0.632) is used here. The predicted masks are refined, as described in Sec. A.3.3. The results of Table A.3 show that GrabCut improves the performance significantly. Both, the Plant IoU as well as the BF-score have improved by approximately 7%. On the contrary, the marker-based watershed is shown to have a negative effect on the overall pipeline performance.

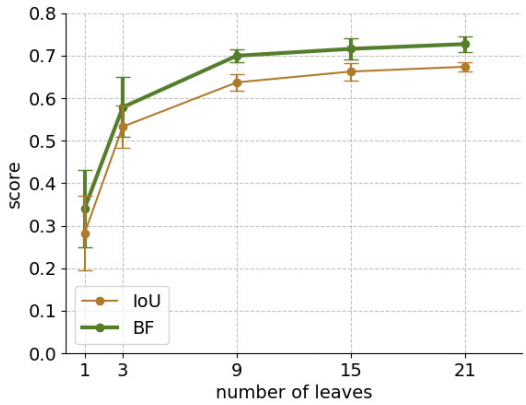


Figure A.6: Mean Plant IoU and BF score (each averaged over 10 iterations) for models trained on synthetic images generated with various numbers of randomly drawn leaves.

Table A.3: Performance results for the mask refinement step using GrabCut or watershed, compared to using no refinement.

Post-processing method	GAcc	Plant IoU	BF
None	0.985	0.632	0.685
GrabCut	**0.988**	**0.703**	**0.751**
Watershed	0.978	0.559	0.567

In Fig. A.7, we illustrate qualitative results using GrabCut in the mask refinement. Even if the model predictions tend to under-segment the detected leaves (second column), those under-segmentations are often corrected by GrabCut (third column), with the trade-off of losing small-scale predictions, like e.g. in the second row.

A.5.4 Generalizability over different fields, days, and plants

In this last experiment, the final pipeline is applied to the test splits of all four data collections (dc) to evaluate the generalization capabilities. As an outcome of the three previous experiments, our final pipeline (i) applies Mix2 image blending, (ii) randomly draws 9 segmented leaves, and (iii) uses GrabCut for post-processing.

Table A.4 contains the obtained experimental results and shows clearly that the Few-leaf training pipeline generalizes over different days (dc1 vs. dc2 vs. dc3), differ-

Figure A.7: Qualitative examples of model predictions without and with GrabCut refinement. GrabCut generally improves large detections but suppresses small ones.

ent fields (dc1 vs. dc2 vs. dc3) and different plants (dc1/dc2/dc3 vs. dc4). However, the performance of data collection 2 is noticeably worse. This is due to the fact that it contains smaller plants, with an average leaf size significantly smaller compared to the average leaf size of data collection 1 and 3 (see again Table C.1).

The proposed Few-leaf training becomes especially relevant when it comes to the creation of high-quality grassland datasets. We picture it as a tool, that can be applied after each data collection session to generate labels for the majority of plant occurrences. The intention is to reduce the manual labeling labor, leaving the human annotator with only a few additional annotations and corrections. We want to get an intuition of how big is the proportion of sufficiently predicted weeds, respectively of how much work remains for the human annotator. We determine for each predicted mask m_p^i corresponding to image i the subset of its ground truth instances S_i that

Table A.4: Performance of the Few-leaf training pipeline on the four different data collections, specified in Table C.1.

Data Collection	GAcc	Plant IoU	BF
1	0.986	0.701	0.704
2	0.991	0.585	0.592
3	0.989	0.745	0.726
4	0.989	0.723	0.729

results in the maximum IoU between m_p^i and S_i. We refer to this maximum value as $IoU_{max,i}$. As a consequence, we know for each ground truth instance the IoU it has been predicted with; it is $IoU = IoU_{max,i}$. Finally, we can determine the overall percentage of ground truth weed pixels in the whole dataset that are predicted with a certain IoU. In Fig. A.8, we show the ratio of ground truth pixel (named *"plant pixel ratio"*) that has been predicted within given ranges of IoUs. The predominant portion of pixels is predicted with an IoU > 0.6 and depending on which label precision is targeted, one can decide if Few-leaf Learning could support during the process of annotating.

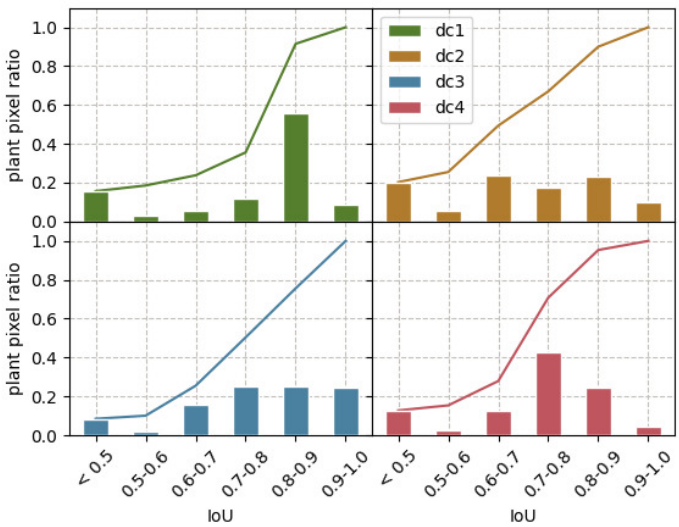


Figure A.8: *Plant pixel ratio* over IoU for the test sets of the four different data collections (dc1 - dc4). The *plant pixel ratio* represents the ratio of ground truth pixels that have been predicted within the pre-defined IoU ranges, shown by the bars. The continuous line represents the cumulative *plant pixel ratio*.

A.6 Conclusion

Our work manifests how domain knowledge—in our case plant phenotyping observations—can lead to meaningful simplifications of demanding computer vision problems. More precisely, we were able to use just 9 random leaves to synthetically generate multiple images and ultimately train accurate weed segmentation models. The mask predictions are further refined by applying GrabCut in an automated way. The proposed pipeline has been evaluated on different days, in different grassland fields, and for two different weeds (Rumex obtusifolius, Cirisium vulgare) exhibiting good generalization capabilities. The presented results show that a significant amount of effort can be saved during annotation using Few-leaf Learning. Hopefully, this will allow for the creation of new and more detailed grassland weed datasets that will enable autonomous robotic weeding. Nevertheless, we believe that our pipeline can be also applied to other domains by properly adapting the domain-specific composition procedure.

A.7 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258.

References

[Ahm+14] Faisal Ahmed et al. "Automated weed classification with local pattern-based texture descriptors." In: *Int. Arab J. Inf. Technol.* 11 (2014), pages 87–94.

[And+20] Marcin Andrychowicz et al. "Learning dexterous in-hand manipulation." In: *The International Journal of Robotics Research* 39.1 (2020), pages 3–20. DOI: 10.1177/0278364919887447.

[BB17] Maxim Berman and Matthew B. Blaschko. "Optimization of the Jaccard index for image segmentation with the Lovász hinge." In: *CoRR* abs/1705.08790 (2017).

[BWS19] Asher Bender, Brett Whelan, and Salah Sukkarieh. "A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica." In: *Journal of Field Robotics* 37 (July 2019). DOI: 10.1002/rob.21877.

[Che+17] Nived Chebrolu et al. "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields." In: *The International Journal of Robotics Research* (2017). DOI: 10.1177/0278364917720510.

[CL13] Gabriela Csurka and Diane Larlus. "What is a good evaluation measure for semantic segmentation?" In: *IEEE Trans. Pattern Anal. Mach. Intell.* Volume 26. January 2013. DOI: 10.5244/C.27.32.

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pages 248–255.

[Di +17] Maurilio Di Cicco et al. "Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection." In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017.

[DMH17] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection." In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob (2017), pages 1310–1319. ISSN: 15505499. DOI: 10.1109/ICCV.2017.146.

[Eve+09] F K van Evert et al. "Real-time vision-based detection of Rumex obtusifolius in grassland." In: *Weed Research* 49.2 (2009), pages 164–174. DOI: https://doi.org/10.1111/j.1365-3180.2008.00682.x.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 770–778. DOI: 10.1109/CVPR.2016.90.

[HO15] Sebastian Haug and Jörn Ostermann. "A Crop/Weed Field Image Dataset for the Evaluation of Computer Vision Based Precision Agriculture Tasks." In: *Computer Vision - ECCV 2014 Workshops*. 2015, pages 105–116. DOI: 10.1007/978-3-319-16220-1_8.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *CoRR* abs/1412.6980 (2015).

[KTN16] T. Kounalakis, G. Triantafyllidis, and L. Nalpantidis. "Weed recognition framework for robotic precision farming." In: *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*. 2016, pages 466–471.

[KTN18] Tsampikos Kounalakis, Georgios Triantafyllidis, and Lazaros Nalpantidis. "Image-based recognition framework for robotic weed control systems." In: *Multimedia Tools and Applications* 77.8 (2018), pages 9567–9594.

[KTN19] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Deep learning-based visual recognition of rumex for robotic precision farming." English. In: *Computers and Electronics in Agriculture* 165 (October 2019). ISSN: 0168-1699. DOI: 10.1016/j.compag.2019.104973.

[Kuz+19] Dmitry Kuznichov et al. "Data augmentation for leaf segmentation and counting tasks in rosette plants." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2019-June (2019), pages 2580–2589. ISSN: 21607516. DOI: 10.1109/CVPRW.2019.00314.

[Man+18] K. Maninis et al. "Deep Extreme Cut: From Extreme Points to Object Segmentation." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pages 616–625.

[Mey92] F. Meyer. "Color image segmentation." In: *1992 International Conference on Image Processing and its Applications*. 1992, pages 303–306.

[Ols+19] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning." In: *Scientific Reports* 9.2058 (1 February 2019). DOI: 10.1038/s41598-018-38343-3.

[Pap+16] Dim P. Papadopoulos et al. "We Don't Need No Bounding-Boxes: Training Object Class Detectors Using Only Human Verification." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pages 854–863.

[PGB03] Patrick Pérez, Michel Gangnet, and Andrew Blake. "Poisson Image Editing." In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. San Diego, California: Association for Computing Machinery, 2003, pages 313–318. ISBN: 1581137095. DOI: 10.1145/1201775.882269.

[Pol+07] G. Polder et al. "Weed Detection Using Textural Image Analysis." English. In: *EFITA/WCCA Conference*. 2007.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ""GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts." In: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. Los Angeles, California: Association for Computing Machinery, 2004, pages 309–314. ISBN: 9781450378239. DOI: 10.1145/1186562.1015720.

[RLF15] O. Russakovsky, L. Li, and L. Fei-Fei. "Best of both worlds: Human-machine collaboration for object annotation." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pages 2121–2131. DOI: 10.1109/CVPR.2015.7298824.

[Shr+17] Ashish Shrivastava et al. "Learning from Simulated and Unsupervised Images through Adversarial Training." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pages 2242–2251. DOI: 10.1109/CVPR.2017.241.

[Sko+19] Soren Skovsen et al. "The grassclover image dataset for semantic and hierarchical species understanding in agriculture." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2019-June (2019), pages 2676–2684. ISSN: 21607516. DOI: 10.1109/CVPRW.2019.00325.

[Sof+20] Konstantin Sofiiuk et al. "f-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation." In: *arXiv preprint arXiv:2001.10331* (2020).

[Tob+17] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world." In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pages 23–30. DOI: 10.1109/IROS.2017.8202133.

[Tod+20] Yosuke Toda et al. "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping." In: *Communications Biology* 3 (April 2020), page 173. DOI: 10.1038/s42003-020-0905-5.

[Tri+19] Shashank Tripathi et al. "Learning to generate synthetic data via compositing." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pages 461–470.

[Tsa+18] Yi-Hsuan Tsai et al. "Learning to Adapt Structured Output Space for Semantic Segmentation." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pages 7472–7481. DOI: 10.1109/CVPR.2018.00780.

[WMH18] Daniel Ward, Peyman Moghadam, and N. Hudson. "Deep Leaf Segmentation Using Synthetic Data." In: *BMVC*. 2018.

[Zha+17] Hengshuang Zhao et al. "Pyramid Scene Parsing Network." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pages 6230–6239. DOI: 10.1109/CVPR.2017.660.

[Zha+20] Shiyin Zhang et al. "Interactive Object Segmentation With Inside-Outside Guidance." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Zhu+17] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

PAPER B

Self-supervised contrastive learning on agricultural images

Ronja Güldenring[1,*] and Lazaros Nalpantidis[1]

*Corresponding author: ronjag@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

Abstract

Agriculture emerges as a prominent application domain for advanced computer vision algorithms. As much as deep learning approaches can help solve problems such as plant detection, they rely on the availability of large amounts of annotated images for training. However, relevant agricultural datasets are scarce and at the same time, generic well-established image datasets such as ImageNet do not necessarily capture the characteristics of agricultural environments. This observation has motivated us to explore the applicability of self-supervised contrastive learning on agricultural images. Our approach considers numerous non-annotated agricultural images, which are easy to obtain, and uses them to pre-train deep neural networks. We then require only a limited number of annotated images to fine-tune those networks in a supervised training manner for relevant downstream tasks, such as plant classification or segmentation. To the best of our knowledge, contrastive self-supervised learning has not been explored before in the area of agricultural images. Our results reveal that it outperforms conventional deep learning approaches in classification downstream tasks, especially for small amounts of available annotated training images where up to 14% increase of average top-1 classification accuracy has been observed. Furthermore, the computational cost for generating data-specific pre-trained weights is fairly low, allowing one to generate easily new pre-trained weights for any custom model architecture or task.

B.1 Introduction

In agriculture, there are only a few publicly available datasets that are large enough to be suitable for state-of-the-art deep learning approaches, as such approaches generally require large amounts of training images. Nevertheless, the required dataset size is proportional to the number of plant species considered in it. Datasets such as DeepWeeds [Ols+19] or Sugar Beets 2016 [Che+17] are common choices for evaluating newly proposed computer vision approaches in the domain of agriculture, but they only consider a relatively small number of species. Yet, from a practical point of view, the introduction of vision-equipped robots in different precision agriculture tasks requires consideration of many different plant species. As is the trend with other domains, such as e.g. medical imaging [Azi+21], where annotated training examples are difficult to obtain, label-efficiency gains stronger attention [Arg+20; Gül+21; Mad+19; Sko+19]. Label-efficient training requires fewer manually annotated training examples, allowing one to easily integrate new plant species for custom robot applications.

One way to achieve label efficiency is semi-supervised training with self-supervision. The key idea of self-supervised pre-training procedures is to generate automatic labels for unlabeled data; then, a deep neural network can be trained in a supervised manner with these labels. Although the formulated task in self-supervised pre-training is generally not meaningful, the network still learns relevant features of the given data. In a second step, the obtained network weights can be transferred to relevant downstream tasks, such as object detection or segmentation. Recently, self-supervised training using contrastive learning received strong attention within the domain of Deep Learning in Computer Vision. Contrastive approaches such as SimCLR [Che+20] and SwAV [Car+20] achieved comparable results to the well-known pre-trained ImageNet weights, even if the latter are obtained in a fully supervised manner.

In this work, we explore the applicability of contrastive learning in the area of agriculture, motivated by the fact that well-established and widely-used ImageNet weights are obtained with image data that differs semantically as well as in texture and color from agricultural images. To the best of our knowledge, contrastive learning has not yet been explored in the area of agricultural images.

More specifically, we compare supervised pre-training from ImageNet with self-supervised training on agriculture-specific data. We use SwAV to obtain pre-trained weights for three different datasets: (1) DeepWeeds [Ols+19], (2) Aerial Farmland [Chi+20], and (3) Grassland Europe. The latter is generated by scraping 24.000 grassland images from the web, which is a simple way of collecting plant image data. The dataset-specific pre-trained weights are compared to supervised ImageNet weights by transferring them to classification and segmentation downstream tasks on the different considered datasets. Figure E.1 shows the *average top-1 accuracy*, trained on different fractions of the DeepWeeds dataset, demonstrating the label efficiency of self-supervised models. Furthermore, we find it beneficial that self-supervised training allows one to easily generate pre-trained weights for any network architecture. We

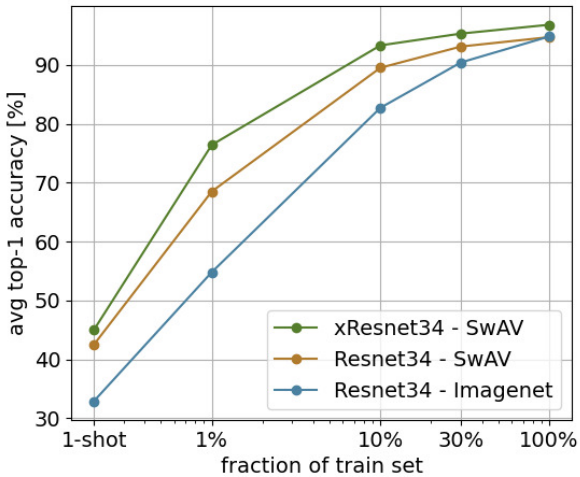


Figure B.1: Average top-1 classification performance for models trained on different fractions of the DeepWeeds dataset. We apply transfer learning and compare supervised ImageNet training to self-supervised pre-training using SwAV. With an increase of average top-1 accuracy up to 14%, SwAV is particularly efficient in semi-supervised settings, where only a fraction of image annotations are needed. Furthermore, SwAV allows us to easily obtain pre-trained weights for any network structure. Exemplary, we show that minimal network modifications such as concat pooling and xResNet lead to a further performance increase.

demonstrate that minimal network modifications increase the top-1 accuracy even further, whereas pre-trained ImageNet weights are not available for custom network designs.

Our key findings can be summarized in the following:

- Self-supervised models predominantly improve the classification performance compared to supervised models. They are especially effective in semi-supervised settings, where we performed fine-tuning on fractions of the datasets.

- Inspired by Azizi et al. [Azi+21], we show that initializing the self-supervised model with supervised ImageNet pre-trained weights can lead to further minimal improvement, but also speeds up the convergence of the self-training.

- The computational cost of obtaining data-specific pre-trained weights with SwAV is relatively low and allows one to quickly obtain weights for custom model architectures. Exemplary, we show that experimenting with minimal network modification – including concat pooling and xResnet modifications – can boost classification performance significantly.

- In our experiments, we show that self-supervised models are capable of dealing successfully with a variety of downstream tasks, but are less effective in segmentation tasks.

The structure of this research work is as follows: Section C.2 gives an overview of relevant literature for advances in Unsupervised Representation Learning and Representation Learning in the context of Agriculture. Section B.3 presents the methodology and architectural design choices in great detail. Section E.5 presents the considered datasets for evaluation and defines the performed experiments, while these experiments are presented and discussed individually in Section B.5. Finally, Section B.6 concludes our work and points out potential future work.

B.2 Related Work

B.2.1 Unsupervised Representation Learning

The motivation behind *Unsupervised Representation Learning*, also known as *Self-supervised Learning*, is to learn feature representation with unlabeled data only. This process is referred to as *pre-training* and it is commonly implemented by formulating a so-called *pre-text task*: a supervised task formulation where labels can be obtained automatically. The pre-text task has no necessary relation to the final network usage but indirectly forces the network to learn relevant low- and high-level features. Common practice is to further use the pre-trained feature extractor and fine-tune it in a *downstream task*, where the final task—such as classification, object detection or segmentation—will be learned in a supervised manner. Optimally, the network has already learned all relevant features during the pre-text task and only needs to learn the relation between high-level features and label names. Usually, self-supervised pre-training allows the downstream task to converge faster and improves label efficiency. [JT21]

A selection of methods for generating pre-text tasks is listed in the following:

- Transformation-based methods. They are based on the assumption that image transformations do not change the semantic content of an image. Therefore, an image along with all its possible image transformations can be treated as one class [Dos+14; GSK18].

- Patch-based methods. The task is to predict the relative position between patches originating from the same image [DGE15; NF16].

- Generation-based methods. The task is to let the network generate information, that is already available and will be used as ground truth, such as image colorization [ZIE16] or image patch reconstruction [Pat+16].

- Temporal-based methods. The task is to predict the temporal order of images extracted from a video [MZH16].

- Cluster-based methods. Feature vectors extracted during the forward pass of a network are further processed by cluster algorithms. The cluster assignments serve as pseudo labels for the loss computation and backward pass [Car+18; YPB16].

Very recently, self-supervised Learning using Noise Contrastive Estimation (NCE) [GH10; OLV18] has gained attention, as it provides results comparable to state-of-the-art supervised methods trained on ImageNet [Den+09]. Instance discrimination methods apply the concept of contrastive learning to whole images. Common practice is to use image transformations to generate different image views, while image views originating from the same image form positive image pairs, and those originating from different images form negative pairs. The objective of NCE is to push positive pairs close to one another and negative pairs further apart.

Google Research has proposed a relatively simple framework: SimCLR [Che+20]. SimCLR processes image batches through a simple encoder network, while for each image pair in the batch, all other samples are treated as negative pairs. In their experiments, they show that contrastive learning benefits from large batch sizes ($\approx$4096) and long training epochs. However, this requires extremely much computational capacity and is therefore not feasible for everyone.

In contrast to SimCLR, MoCo [He+20] from Facebook AI Research is more applicable for training on limited hardware. Instead of large batch sizes, they introduce a queue that keeps negative examples from previous batches. Furthermore, they introduce a momentum encoder for encoding negative examples that has the same architecture as the original encoder network. During training, only the encoder is updated while weights are copied to the momentum encoder with a certain time delay, i.e. the momentum encoder is behind the encoder.

While MoCo and SimCLR stress the importance of a large number of negative samples, BYOL [Gri+20] and SimSiam [CH21] show that state-of-the-art performance can be achieved without providing any negative examples actively. Caron et. al. proposes SwAV [Car+20], which combines concepts of contrastive learning with cluster-based learning methods. The main advantage of SwAV is that it allows negative and positive examples to be similar, which can be easily the case when two different images are semantically similar. Furthermore, SwAV computes cluster assignments online and works with relatively small batch sizes. SwAV is explained in more detail in Section B.3.2 as it constitutes an integral part of our work presented in this paper.

Table B.1 provides an overview of the most recent publications in the field of self-supervised contrastive learning and lists the key design choices of each approach.

All presented self-supervised contrastive learning approaches have been applied to the benchmark dataset ImageNet which includes more than 14 million images of diverse classes. But how well do those methods work in more specific domains with substantially smaller dataset sizes? Azizi et al. [Azi+21] recently applied contrastive learning to the domain of medical images, considering classification datasets containing “only” ten to hundred thousand of images. They show, that classic supervised ImageNet transfer learning can be outperformed with domain-specific self-supervision.

Table B.1: State-of-the-art self-supervised contrastive learning approaches with its key design choices.

Self-supervised Contrastive Learning Architecture	Key Design Choices
SimCLR [Che+20]	• Simple Encoder Network. • InfoNCE loss [OLV18], considering negative and positive pairs. • Extreme large batch sizes with many negative samples (≈4096).
MoCo [He+20]	• Encoder and momentum encoder network, which is a slow-moving average of the encoder network and encodes the negative samples. • InfoNCE loss [OLV18], considering negative and positive pairs. • Queue with negative samples from previous batches which allows smaller batch sizes.
BYOL [Gri+20]	• Online and target network, which is a slow-moving average of the online network. Each network predicts one view of an image. • Loss is the mean-square error of both network predictions. No contrasting towards negative samples, only positive pairs are considered.
SwAV [Car+20]	• Encoder network followed by feature vector clustering. • Cross-entropy loss function on swapped predictions, considering negative and positive pairs. • SwAV works with small batch sizes and allows negative and positive examples to be similar. • Introduction of an effective data augmentation method *multi-crop*, which is also beneficial in the other approaches.
SimSiam [CH21]	• Encoder network 1 combined with a projection head (MLP) and encoder network 2 without a projection head. Both encoder networks share weights. Only the first network gets updated. • Loss is the negative cosine similarity. No contrasting towards negative samples, only positive pairs are considered.

However, they achieve even better performance when combining pre-training both on ImageNet and medical images.

B.2.2 Agricultural Representation Learning

Although agricultural images provide significant characteristic differences compared to ImageNet data [Den+09], it is a common practice to use ImageNet weights for transfer learning. A number of publications in the domain of agriculture, e.g. Kounalakis er al. [KTN19], Mehdipour Ghazi et al. [MYA17] and Suh et al. [Suh+18], show that ImageNet pre-trained weights improve the model performance. However, little research has been performed on whether more domain-specific transfer learning could improve performance even further. Bargoti and Underwood[BU17] transfer weights between fruit orchards, but benefits over ImageNet weights are only shown when training with very few images. Bosilij et al. [Bos+20] train supervised on one specific crop and transfer these weights to fine-tune on another crop. They show that initializing networks with similar-domain weights leads to more efficient training and only requires a fraction of the ground truth labels to achieve similar performance. Espejo-Garcia et al. [Esp+20a] generate agricultural pre-trained weights on the highly diverse, but relatively small *Plant Seedlings Dataset* [MYA17] and test them on the more specific *Early Crop Weeds Dataset* [Esp+20b]. They perform experiments for four popular CNN architectures and report an overall slight improvement in performance and reduction in training convergence time. Marin Zapata et al.[Mar+20] learn a representation of a plant phenotyping dataset with a triplet network architecture, inspired by Wang et al. [Wan+14]. They take advantage of the underlying timing sequence of the images in their dataset, where consecutive image frames are treated as similar and non-consecutive image frames as dissimilar. The learned representation is used in various downstream tasks such as clustering and classification. Zhao et al. [Zha+20] propose a multi-task learning framework that combines supervised classification with a hand-crafted self-supervision task, inspired by Gidaris et al[GSK18]. The purpose of the self-supervision component is to speed up the optimization by setting a focus on the encoding of image orientation. dos Santos Ferreira et al. [dos+19] applied two unsupervised deep clustering algorithms JULE [YPB16] and DeepCluster [Car+18] to two agricultural datasets. Unfortunately, they do not transfer the obtained weights but use the trained clustering network to semi-automate the image annotation and re-train a classification model with ImageNet weight initialization.

In contrast to the works discussed above, we use self-supervised contrastive learning to obtain agriculture-specific pre-trained weights. Unsupervised learning is especially relevant in agriculture because collecting images is relatively easy while their manual annotation requires a lot of additional effort. To the best of our knowledge, this is the first work that investigates self-supervised contrastive learning in the domain of agriculture.

B.3 Methods

B.3.1 xResNet architecture

In this work, we will consider two backbone architectures: ResNet-34 and xResNet-34. While the default ResNet architectures are widely known, the xResNet introduces small modifications in the input stem and downsampling block proposed by He et al. [He+19]. Note, that the name *xResNet* was not originally given by its authors He et al. [He+19], but was established by the FastAI community [1]. The (7×7) convolution in the input stem of the ResNet is replaced with three (3×3) convolutions. In the downsampling block, two modifications are applied: (1) in path A, the stride is applied in the second convolution instead of the first, (2) in path B, a (2×2) average pooling with stride 2 precedes the convolutional layer, whose stride is reduced to 1. The modifications are illustrated in Figure B.2.

Such small modifications come with a significant cost: To apply transfer learning, new pre-trained weights must be obtained, as the ones for ResNet are not applicable. This is usually done with ImageNet [Den+09], which consists of ≈ 14 million images, resulting in computationally expensive training. In this work, we do not have ImageNet pre-trained weights available because of computational limitations. Instead, we will obtain weights using SwAV [Car+20] for self-supervised pre-training.

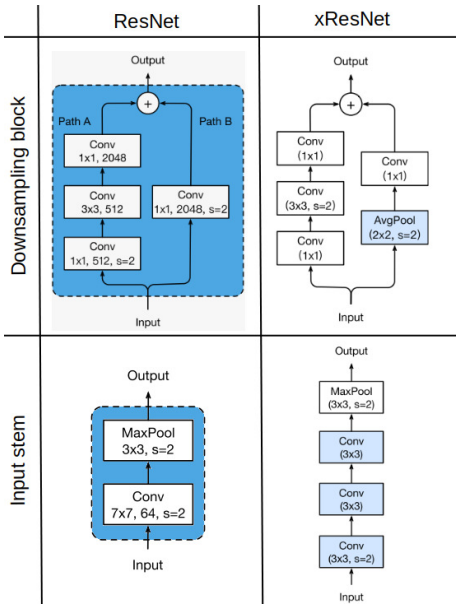


Figure B.2: The xResNet architecture proposed by He et al. [He+19] (right) modifies the original ResNet (left) in the downsampling block and the input stem.

[1] https://docs.fast.ai/vision.models.xresnet.html

B.3.2 SwAV

The work of Caron et al. [Car+20] introduced SwAV, which combines cluster-based learning approaches with contrastive learning. Figure B.3 illustrates the architecture of SwAV, described in the following. A batch X of images $x_1, ..., x_n$ is augmented twice to obtain batches X_1, X_2. Both augmented batches are processed by the same encoder network to obtain corresponding feature vectors Z_1, Z_2. These features are assigned to clusters Q_1, Q_2 by mapping Z_1, Z_2 to k learnable prototypes, which are further processed by the Sinkhorn Knopp algorithm [Cut13]. For the computation of the loss, as shown in Equations B.1 - B.3, let's assume one image x_i from X with two augmented views x_{i1}, x_{i2}. The obtained cluster assignments q_{i1}, q_{i2} serve as labels for the images. The trick hereby is to swap the assigned clusters between the image views, i.e. q_{i1} serves as a target for x_{i2} and vice-versa (Equation B.1). Concretely, the loss function is a cross-entropy loss of probability of the predictions z_{i1}, z_{i2} and the swapped targets q_{i2}, q_{i1} (Equation B.2). The probability of the predictions is computed by taking the Softmax of the vector similarity between the feature vector z_i and the prototype vectors C (Equation B.3).

$$L(z_{i1}, z_{i2}) = l(z_{i1}, q_{i2}) + l(z_{i2}, q_{i1}) \tag{B.1}$$

$$l(z_i, q_i) = -\sum_k q_i^{(k)} \log p_i^{(k)} \tag{B.2}$$

$$\text{where } p_i^{(k)} = \frac{\exp(z_i^T c_k/\tau)}{\sum_{k'} \exp(z_i^T c_{k'}/\tau)} \tag{B.3}$$

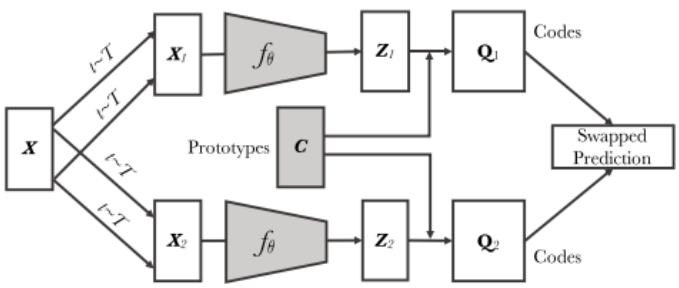


Figure B.3: Architecture of SwAV [Car+20]: An image batch X is augmented twice to obtain (X_1, X_2) which are then processed by the identical feature extractor f_θ. The obtained feature vectors (Z_1, Z_2) are mapped to prototype vectors that are further clustered to (Q_1, Q_2). The clusters serve as pseudo labels and are swapped between the two streams so that the loss is computed between Z_1 and Q_2 and vice versa.

The advantage of SwAV over other recent and successful contrastive learning approaches is, that it allows different images with similar features to form a cluster,

therefore being close to each other. This makes SwAV especially applicable to the agricultural image domain. Compared to ImageNet, agricultural datasets contain relatively similar images. E.g. two different images from the same plant species in a similar growing state should be close, while images of different plant species or different growing stages provide less similar features.

B.4 Experimental Setup

Since we can not directly measure the quality of pre-trained weights, we evaluate their quality by applying downstream tasks to differently initialized models: *Kaiming* initialization [He+15], supervised *ImageNet* pre-trained weights and weights obtained with unsupervised representation learning using SwAV. Inspired by Azizi et al. [Azi+21], we actually consider two self-training protocols: *SwAV* indicates representation learning from scratch, while in *ImageNet* $\rightarrow$ *SwAV* we first load the supervised ImageNet weights and apply self-supervised training with the targeted dataset on top. In the following subsection, the setup of the different experiments (Section B.4.2.1 - B.4.2.2) and datasets (Section B.4.4) is specified in more detail. Furthermore, in each of our conducted experiments, we list the concrete hyper-parameters we used, which in all cases were determined using a grid search approach.

B.4.1 Unsupervised representation Learning with SwAV

The representation is obtained using SwAV as introduced in section B.3.2. The encoder network (either ResNet-34 or xResNet-34) is followed by a pooling layer and a 2-layer multi-layer perceptron (MLP) with a hidden layer of size 256. We perform experiments with *average pooling* and *concat pooling*, that concatenate the output of average and max pooling. The MLP maps the feature vectors to 1000 prototype vectors. We use an image input size of $[244 \times 244]$ and a batch size of 128, that is the maximum we can fit on our Nvidia Tesla V100. Due to the relatively small batch size, we apply a queue of size 960 after 50 epochs for xResnet-34 and after 100 epochs for Resnet-34, containing feature vectors from previous iterations. An earlier introduction of the queue was found to lead to a collapsed training. In total, we train 600 epochs. The network is updated with the ranger optimizer[2] and a constant learning rate of 0.01. The following augmentations are applied to the images: multi-crop [Car+20], scaling, color jitter, grayscale transform, Gaussian blur, and random rotation. Weights are either initialized with Kaiming initialization or with supervised ImageNet weights.

[2] https://lessw.medium.com/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d

B.4.2 Downstream Tasks

We considered two relevant downstream tasks—Classification and Segmentation—for our experiments.

B.4.2.1 Classification

In this work, we investigate whether agricultural domain-specific pre-trained weights can improve label efficiency. Thus, our goal is not to maximize classification performance as such. Therefore, all our experiments on the classification downstream task are performed with the same set of parameters. The encoder network (either ResNet-34 or xResNet-34) is followed by a classification head, consisting of a pooling layer and a linear layer that projects the feature vectors to final class predictions. We use an input image size of $[244 \times 244]$ and train each model for 50 epochs with a batch size of 32. We use Stochastic Gradient Descent (SGD) as an optimizer and cross-entropy loss as the criterion. For the pooling layer in the network head, we perform experiments with *average pooling* or *concat pooling*. We differentiate between two experiments:

- **Linear Evaluation**. It is a measurement of how well the representation is learned. During training, only the weights of the head are updated, while the encoder weights are kept frozen. The learning rate is set to $lr_head = 0.01$ and decayed twice at epoch [28, 41] with $\gamma = 0.2$. Augmentations similar to the DeepWeeds baseline in [Ols+19] are applied during training: random scale, random crop, random rotation, RGB shift, and horizontal and vertical flip.

- **Classification Fine-tuning**. The model is fine-tuned with an initial learning rate of $lr_enc = 0.001$ for the encoder and $lr_head = 0.01$ for the head. These learning rates are decayed twice at epoch [28, 41] with $\gamma = 0.2$. The same augmentations as in the *Linear Evaluation Model* are applied.

For all our experiments the *average top-1 accuracy* is reported, i.e. for each class the top-1 accuracy is obtained separately and finally, the sum is averaged over all classes.

B.4.2.2 Segmentation Fine-tuning

We also perform experiments on segmentation downstream tasks, which are—compared to classification downstream tasks—more complex and less related to instance discrimination methods such as SwAV. Again, we don't aim for performance maximization but rather investigate whether transferring weights obtained with SwAV is beneficial in the agriculture domain. Therefore, we apply the widely known UNet [RPB15] with a ResNet-34 as the encoder. We use the Adam optimizer [KB15] with a cosine annealing [LH17] as learning rate scheduler, while the initial learning rates are 0.0001 for the encoder and 0.001 for the decoder. As a criterion we apply Lovasz Loss [BRB18]. For all segmentation experiments, we report the *mean Intersection over Union (mIoU)*

– *also called Jaccard Index* – *[Jac12]* on the test set. For each class, a separate IoU is computed by dividing the number of intersecting pixels between prediction and ground truth with the sum of all pixels presented in both. The mIoU is the averaged sum over all classes.

B.4.3 Similarity of Neural Network Representation

Neural network representations can be compared using Linear Centered Kernel Alignment (CKA) [CMR12; Cri+02], as proposed by [Kor+19]. Thereby, the similarity between layers of two representations is evaluated. To get a better understanding of what is learned with SwAV, we perform the Linear CKA analysis on representations learned with the DeepWeeds Dataset. We use the code provided by [Kor+19] and compute the similarity for the output of each ResNet-34 block, resulting in a similarity matrix of 17×17 blocks. For the similarity computation, we consider 100 randomly drawn test images.

B.4.4 Datasets

In our work, we benchmark the proposed method on publicly available large datasets such as *DeepWeeds* [Ols+19] and *Aerial Farmland* [Chi+20]. Additionally, we apply the method to a more practical scenario where we collect a relatively small and platform-specific dataset *Rumex Denmark 2020* of the Grassland weed Rumex obtusifolius. Then, we enrich the data from *Rumex Denmark 2020* further with images from different Grassland species scraped from the internet, which we call *Grassland Europe*.

B.4.4.1 DeepWeeds

DeepWeeds [Ols+19] is an image classification dataset of weed species from Australian rangelands. In total, it contains 9 different classes—8 weed species shown in Figure B.4 and a background class. It provides 17,509 RGB images of size $[256 \times 256]$, while the number of negative images predominates with a share of 52%. We split the dataset randomly in 60% training–, 20% validation– and 20 % test–split. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.
- Linear Evaluation: We balance the training data by under-sampling all classes to the smallest number, which is represented by the species *Snake Weed*. The balanced images are used to obtain the results for the linear evaluation model. Results are reported on the raw test–split.
- Classification Fine-tuning: We fine-tune the model on the balanced training data as described previously. The model is fine-tuned on the whole balanced

Figure B.4: One sample image per species of the DeepWeeds dataset [Ols+19]. Top row: chinee apple, lantana, parkinsonia, parthenium; bottom row: prickly acacia, rubber vine, siam weed, snake weed

training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%, 0.2%). For each fraction, we draw 5 different folds and report the mean and standard deviation over the 5 folds. Note, that 0.2% is equal to one-shot learning with one image per class. Results are reported on the raw test–split.

B.4.4.2 Aerial Farmland

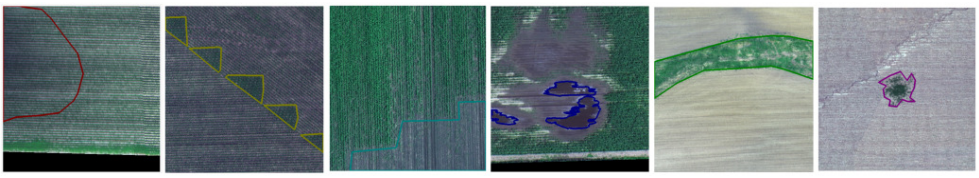


Figure B.5: One sample image per anomaly class of the Aerial Farmland dataset [Chi+20]. From left to right: cloud shadow, double plant, plant skip, standing water, waterway, weed cluster.

We consider the publicly available challenge dataset[3] for semantic segmentation of six different anomaly patterns, that is a subset of the Aerial Farmland Dataset [Chi+20]. Example images of the considered anomaly patterns are shown in Figure B.5. Originally, it included 21,061 (train: 12,901, validation: 4,431, test: 3,729) high-quality aerial RGB as well as corresponding near-infra-red (NIR) images from farmlands in the United States of America. Since it was made public within an ongoing challenge, we have no access to the annotations of the test split. Therefore,

[3] https://www.agriculture-vision.com/agriculture-vision-2020/dataset

we split the validation set in two halves serving as validation– and test splits in our experiments. Furthermore, additional binary masks (which we call *valid_mask* in the following) are provided by Chiu et al. [Chi+20], that indicate which areas in the images are valid farmland and which are not, like e.g. streets, rivers or woods. These invalid farmland pixels are marked in black in Figure B.5.

Looking more closely at the data, one can observe that the majority of images contain only a single segmentation class (more precisely train: 11,436, validation: 4,157). This allows us to re-formulate the task as a classification task, where images containing annotations of only one class are considered and labeled with that specific class. The information of the *valid_mask* is incorporated by blacking out the stated invalid pixels. We consider only RGB images of size $[244 \times 244]$. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.
- Linear Evaluation: The dataset is highly imbalanced. In contrast to DeepWeeds, under-sampling is no option because it would reduce the amount of data too heavily. Instead, we weigh the contribution of each class in the cross entropy loss by the inverse of their number of appearances. Results are reported on the test–split.
- Classification Fine-tuning: During fine-tuning, we use the weighted cross-entropy loss as described in the *Linear Evaluation. We evaluate the classification performance leaned on the classification experiments of the DeepWeeds dataset: The model is fine-tuned on the whole training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%), while 5 different folds are drawm and the mean and standard deviation are reported over these 5 folds. For the Aerial Farmland dataset, 1% is the lowest possible fraction because then one of the classes is represented with exactly one image.*
- Segmentation Fine-tuning: As with the other experiments, we use an input image size of $[244 \times 244]$ and train each model for ≈ 12000 iterations with a batch size of 32. The model is fine-tuned on the whole training–split as well as on fractions (10%, 1%) like in previous experiments.

B.4.4.3 Grassland Europe

We created a dataset specifically for applications in the Grasslands of Europe by scraping relevant images from PlantNet[4]. In total, we downloaded 24,665 RGB images of nine common grassland species in Europe: Cirsium Vulgare, Crepis Vesicaria, Glechoma Hederacea, Plantago Major, Rumex Crispus, Rumex Obtusifolius, Ulex Europaeus, Urtica Dioica, Urtica Urens. Example images for each species are shown

[4] Pl@ntNet project: https://identify.plantnet.org/

Figure B.6: One sample image per species of the Grasslands Europe dataset. Top row: Cirsium Vulgare, Crepis Vesicaria, Glechoma Hederacea, Plantage Major, Rumex Crispus; bottom row: Rumex Obtusifolius, Ulex Europaeus, Urtica Dioica, Urtica Urens.

in Figure B.4. All images are resized to 244 pixels on the smaller edge, while keeping the original aspect ratio, followed by a random crop of $[244 \times 244]$. We split the dataset in 60 % training–, 20% validation– and 20% test–split. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.
- Linear Evaluation: Similarly to the Aerial Farmland dataset, we address the class imbalance by weighting each contribution to the cross entropy loss.
- Classification Fine-tuning: Again, the model is fine-tuned on the whole training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%, 0.5%). We report the mean and standard deviation of the average top-1 accuracy over the 5 folds. For the Grassland Europe dataset, 0.5% is the lowest possible fraction because then one of the classes is represented with exactly one image.

B.4.4.4 Rumex Denmark 2020

Figure B.7: Sample images of *Rumex Denmark 2020* dataset from different fields and days with overlayed segmentation mask in orange.

Furthermore, we have collected 282 images of the invasive grassland species *Rumex obtusifolius* and created manual pixel-wise annotations. The images have been taken with a 8-megapixel 1/3-inch camera with a resolution of [2448 × 3263] while freely moving through the field. As a consequence, the collected images contain different levels of blur. Images were collected at four different locations in Denmark on four different days in Spring and Autumn 2020, providing a certain level of variance in lightning, plant as well as background appearance. Example data is shown in Figure B.7.

- Segmentation Fine-tuning: The model is fine-tuned on image sizes of [480 × 640] for 30 epochs and with a batch size of 32. In this experiment, we only consider 100% of the data because the dataset is already relatively small.

B.5 Results and Discussion

Now, we will show and discuss the results for the defined experiments in Section E.5, while table B.2 serves as an overview of which experiments are performed for which datasets.

Table B.2: Overview of performed experiments.

Experiments	Model	Dataset			
		DeepWeeds [Ols+19]	Aerial Farmland [Chi+20]	Grassland Europe	Rumex Denmark 2020
Linear Evaluation	ResNet-34	✓	✓	✓	
Classification Fine-tuning	ResNet-34	✓	✓	✓	
	xResNet-34	✓			
Segmentation Fine-tuning	ResNet-34		✓		✓
Representation Similarity	ResNet-34	✓			

B.5.1 Linear Evaluation

In this section, we examine how well the representation is learned with SwAV for the different datasets. We perform *Linear Evaluation* for the ResNet-34 and compare different weight initialization: Kaiming initialization, supervised ImageNet pre-trained weights and weights obtained with unsupervised representation learning using SwAV. Table B.3 lists the results for all three classification datasets. For all three datasets, the learned representation using SwAV outperforms the ImageNet representation significantly. The results confirm the expectation that more data-specific features are learned. Note, that initializing SwAV with ImageNet weights (ImageNet $\rightarrow$ SwAV) even leads to a slightly better representation. Another advantage of initializing the self-training with ImageNet weights is faster convergence during training as shown in Figure B.8. For all three datasets, the loss drops faster for networks that have been initialized with ImageNet weights (*ImageNet* $\rightarrow$ *SwAV*, green). A possible explanation is that general image features such as edges and corners are learned already and SwAV can directly start learning more data-specific features.

Table B.3: Linear Evaluation Model for the three classification datasets: DeepWeeds, Aerial Farmland and Grassland Europe. During training, only the layers on top of the encoder architecture are updated. We compare the average top-1 accuracy of four weight initializations, fine-tuned on 100% of the data.

	DeepWeeds	Aerial Farmland	Grassland Europe
weight initialization	average top-1 accuracy		
Kaiming	29.5	37.7	20.7
ImageNet	75.3	61.4	71.2
SwAV	94.4	68.4	85.5
ImageNet → SwAV	**94.9**	**70.6**	**86.4**

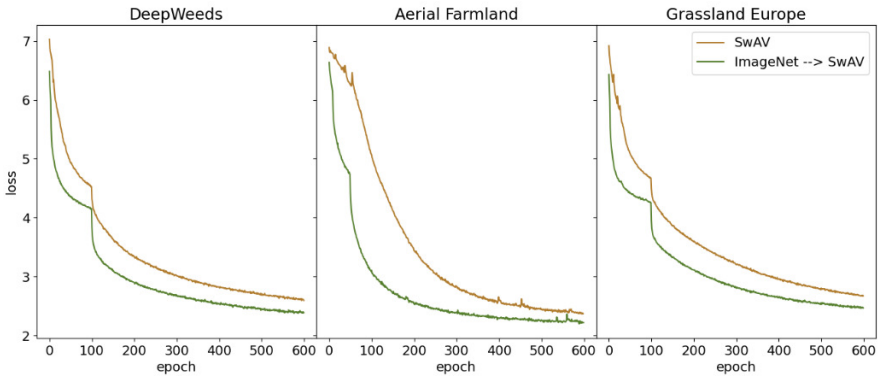


Figure B.8: Training loss curve for training from scratch (*SwAV*, orange) vs. with ImageNet initialization (*ImageNet* → *SwAV*, green) for all three datasets. For all three datasets, the loss drops faster for networks that have been initialized with ImageNet weights. Note, that the knik is caused by the introduction of the queue.

B.5.2 Classification Fine-tuning

In this section, we examine whether our newly obtained pre-trained weights can reduce the amount of labeled data needed for fine-tuning. Our model has been fine-tuned on different fractions of the training dataset for all three classification datasets as shown in Table B.4. Note, that for each dataset the lowest possible fraction is where at least one of the classes is represented with exactly one image. The results show, that SwAV pre-training is especially effective when fine-tuning on less labeled data. For all datasets and all chosen subsets, we get a higher accuracy up to 15% with

SwAV weights. For DeepWeeds, using SwAV pre-training, even with 30% of labeled data, we outperform the classical transfer learning with ImageNet weights and 100% of labeled data. However, for Aerial Farmland and Grassland Europe fine-tuning on all images (100%), leads to a better performance with standard ImageNet weights. All in all, the results show that one can profit highly from SwAV pre-training, when the image data collection is relatively easy, while the labour of manual annotations remains expensive. One would pre-train on all collected images while fine-tuning only on a fraction of labeled images.

Table B.4: Classification performance for different weight initialization and all three agricultural classification datasets. The model is fine-tuned on the complete training dataset (100%) and on randomly drawn subsets. For each dataset the lowest possible fraction is where at least one of the classes is represented with exactly one image, i.e. DeepWeeds: 0.2%, Aerial Farmland: 1%, Grassland Europe 0.5%.

train img fraction [%]	weight init	DeepWeeds	Aerial Farmland	Grassland Europe
		average top-1 accuracy (mean ± std)		
100	Kaiming	73.8 ± 0	65.0 ± 0	73.6 ± 0
	ImageNet	94.3 ± 0	**80.6** ± 0	**91.7** ± 0
	SwAV	95.9 ± 0	72.1 ± 0	88.6 ± 0
	ImageNet → SwAV	**96.1** ± 0	77.2 ± 0	89.9 ± 0
30	Kaiming	51.7 ± 2.0	50.9 ± 2.0	42.8 ± 1.1
	ImageNet	89.9 ± 0.9	67.8 ± 9.3	86.7 ± 0.3
	SwAV	93.5 ± 0.7	70.5 ± 1.1	85.9 ± 0.5
	ImageNet → SwAV	**94.6** ± 0.7	**72.2** ± 1.0	**88.3** ± 0.3
10	Kaiming	40.2 ± 0.8	38.4 ± 2.0	29.4 ± 0.5
	ImageNet	82.3 ± 1.0	57.5 ± 10.6	80.4 ± 0.8
	SwAV	89.4 ± 0.4	**68.4** ± 1.1	**83.3** ± 1.2
	ImageNet → SwAV	**92.0** ± 0.7	68.3 ± 0.8	83.1 ± 7.4
1	Kaiming	25.4 ± 0.6	28.5 ± 1.9	17.9 ± 0.8
	ImageNet	54.0 ± 2.7	50.7 ± 2.9	53.2 ± 2.0
	SwAV	68.0 ± 4.0	**56.6** ± 2.3	67.9 ± 2.7
	ImageNet → SwAV	**69.0** ± 0.3	56.4 ± 2.5	**74.3** ± 2.3
0.5	Kaiming	22.5 ± 2.2		17.3 ± 0.7
	ImageNet	45.6 ± 2.5		46.7 ± 1.1
	SwAV	60.0 ± 2.8		62.7 ± 0.9
	ImageNet → SwAV	**62.8** ± 2.6		**69.7** ± 3.0
0.2	Kaiming	17.7 ± 3.7		
	ImageNet	33.8 ± 4.1		
	SwAV	44.3 ± 4.2		
	ImageNet → SwAV	**46.4** ± 3.6		

Unsupervised Representation Learning is not just a way to generate better weight initialization but also provides flexibility in the design of encoder networks. One doesn't rely anymore on architectures, that come with pre-trained ImageNet weights such as ResNet [He+16], VGG16 [SZ15] or InceptionV3 [Sze+16]. Exemplary, we modified the baseline encoder ResNet-34, obtained new SwAV weights, and fine-tuned the network on the fractions of the DeepWeeds dataset. Two modifications are performed:

- Concat Pooling: We simply switch out the average pooling in the classification head with concat pooling. Since this modification doesn't affect the encoder network directly, we can further compare SwAV weights with ImageNet weights.

- xResNet-34: Additionally, we use a xResNet-34 as described in section B.3.1. ImageNet weights do not exist for the xResNet-34 architecture. Therefore, we only apply SwAV pre-training which is computationally cheaper and therefore feasible to obtain.

Table B.5 lists the average top-1 accuracy for the mentioned modifications as well as the performance difference compared to Table B.4. Interestingly, replacing average pooling with concat pooling improved performance for the SwAV initialization, but reduced the performance heavily for the ImageNet weights. The performance drop can be explained by the fact that pre-trained ImageNet weights have been obtained with average pooling in the classification head, while for the generation of SwAV weights, we directly used concat pooling to adapt the encoder design optimally. Furthermore, the modification that comes with the xResNet architecture improved the average top-1 accuracy even further for all data fractions. With only 30% of the training data, the xResNet-34 outperforms the results obtained with ResNet-34 initialized with ImageNet weights and 100% of the data (cf. Table B.4). These results show that one is not restricted anymore to architectures that come with pre-trained weights ImageNet but can design application-specific architectures and obtain pre-trained weights relatively cheaply using self-supervised contrastive learning. Figure E.1 visually reflects the main results for fine-tuning on DeepWeeds with different weight initialization and network modifications. Note, that the presented model modifications are specifically selected for DeepWeeds and did not show the same effect on the other two datasets.

It is also worth mentioning that the cost of obtaining SwAV weights is lower, compared to the generation of ImageNet pre-trained weights. We performed our SwAV training only on a single Nvidia Tesla V100. The training time scales obviously with the number of images per epoch, but also with the selected augmentations methods; e.g. Gaussian blur is a time-consuming augmentation method that increases the training time of SwAV significantly. The SwAV training for a ResNet-34 on Deepweeds with 10513 training images for 600 epochs lasts 13:19 (h:min)—without Gaussian blur augmentation this can be reduced to 7:19 (h:min).

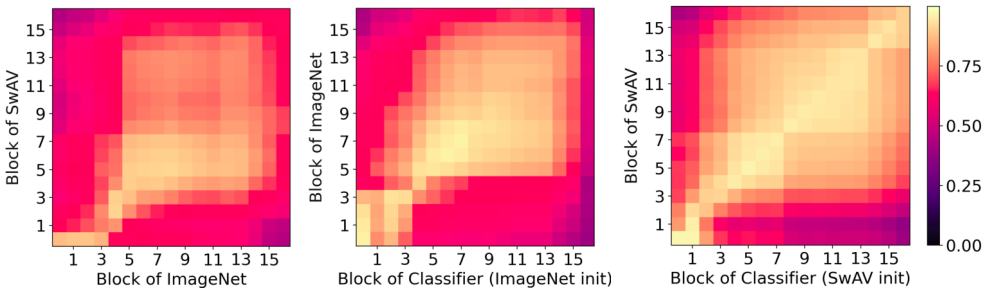
Table B.5: We apply two network modifications: (1) ResNet-34 + concat pooling in the classification head, (2) xResNet-34 modifications, and show their effects on the fine-tuning performance of the balanced training dataset (100%) and training subsets (30%, 10%, 1%, 0.2%) of the DeepWeeds dataset.

train img fraction [%]	arch	weight init	avg top-1 acc (mean ± std)	diff
100	ResNet-34 (+ concat pool)	Kaiming	50.5 ± 0	-23.3
		ImageNet	92.9 ± 0	-1.4
		SwAV	96.3 ± 0	$+0.4$
	xResNet-34	SwAV	$\mathbf{96.6} \pm 0$	$+0.7$
30	ResNet-34 (+ concat pool)	Kaiming	39.0 ± 2.9	-12.7
		ImageNet	84.5 ± 1.5	-5.4
		SwAV	94.4 ± 0.6	$+0.9$
	xResNet-34	SwAV	$\mathbf{95.0} \pm 0.5$	$+1.5$
10	ResNet-34 (+ concat pool)	Kaiming	27.4 ± 3.1	-12.8
		ImageNet	63.7 ± 4.6	-18.6
		SwAV	91.0 ± 0.7	$+1.6$
	xResNet-34	SwAV	$\mathbf{92.6} \pm 0.4$	$+3.2$
1	ResNet-34 (+ concat pool)	Kaiming	19.4 ± 0.9	-6.0
		ImageNet	30.6 ± 5.2	-23.4
		SwAV	74.2 ± 3.0	$+6.2$
	xResNet-34	SwAV	$\mathbf{76.2} \pm 1.6$	$+8.2$
one-shot	ResNet-34 (+ concat pool)	Kaiming	16.9 ± 2.0	-0.8
		ImageNet	28.0 ± 3.8	-5.8
		SwAV	44.7 ± 3.3	$+0.4$
	xResNet-34	SwAV	$\mathbf{45.9} \pm 3.6$	$+1.2$

B.5.3 Similarity of Neural Network Representation for DeepWeeds

In Figure B.9, the similarity of different representations is shown. Figure B.9a shows the similarity of the SwAV and the ImageNet representation. The similarity matrix undermines the expectation, that early blocks ($\approx 0 - 7$) learn more general low-level features and later layers ($\approx 8 - 16$) contain more dataset-specific high-level features that certainly differ between the two representations. In Figure B.9b, B.9c we compare representations of pre-trained weights with final representations of the trained classifier to investigate how much the initial weights have evolved throughout fine-tuning the network. Figure B.9b shows how much ImageNet weights have changed during fine-tuning. There is a high similarity on the diagonal of the first half of the network, meaning early layers have not changed much throughout fine-tuning, while later layers have evolved more strongly. On the contrary, in Figure B.9c the bright

diagonal stretches out through the whole network. The weights of the network initialized with SwAV have not changed much during fine-tuning. It can be concluded, that during self-supervised training relevant data-specific features have been learned for the classification task. These findings are in correspondence with the results from section B.5.1.



(a) **(b)** **(c)**

Figure B.9: Similarity between ResNet-34 blocks for different representations. In (a) SwAV weights are more similar to ImageNet weights in lower layers. Thus, the low-level features learned turn out to be common in both cases. In (b), (c) we compare representations of pre-trained weights (on the vertical axes) with the final representations of trained classifiers (on horizontal axes) to show how much the initial weights have evolved throughout fine-tuning the network. For SwAV (c), the bright diagonal stretches out through the whole similarity matrix, which indicates that low-level, as well as high-level layers, do not change much during fine-tuning. We can conclude, that—contrary to ImageNet (b)—relevant high-level features have been extracted during self-supervised pre-training.

B.5.4 Segmentation Fine-tuning

In this section, we present experiments on two segmentation downstream tasks. The encoder is initialized with transfer learning, while the decoder is always initialized with Kaiming weights. This means, that only half of the network layers are transferred. For the selected segmentation tasks, our data-specific obtained SwAV weights do not perform as convincingly as in classification downstream tasks. More specifically, they do not improve over standard ImageNet weights, not even for very small fractions of training data. For *Aerial Farmland* as shown in Table B.6, models with ImageNet pre-trained weights perform clearly better. For *Rumex Denmark 2020* as shown in Table B.7, the mIoU is very close for ImageNet weights and the different SwAV variants, pre-trained on the two weed datasets *Grassland Europe* and *Deepweeds*.

Segmentation downstream tasks are generally more complex and less similar to instance discrimination methods like SwAV. Since classification is performed on pixel level, different features might be relevant in order to obtain accurate segmentation masks. Therefore, instance discrimination methods like SwAV do not seem to be

suitable for generating data-specific pre-trained weights for more complex downstream tasks.

Table B.6: Segmentation performance on the Aerial Farmland dataset for different weight initialization. The model is fine-tuned on the complete (100%) and on randomly drawn subsets (30%, 10%, 1%) of the Aerial Farmland dataset.

train img fraction [%]	weight init	mIoU (mean ± std)
100	Kaiming	0.296 ± 0
	ImageNet	**0.377** ± 0
	SwAV	0.367 ± 0
	ImageNet → SwAV	0.365 ± 0
10	Kaiming	0.283 ± 0.011
	ImageNet	**0.388** ± 0.005
	SwAV	0.352 ± 0.008
	ImageNet → SwAV	0.370 ± 0.007
1	Kaiming	0.224 ± 0.011
	ImageNet	**0.304** ± 0.019
	SwAV	0.289 ± 0.012
	ImageNet → SwAV	0.293 ± 0.016

Table B.7: Segmentation performance on the Rumex Denmark 2020 dataset for different weight initializations. As SwAV initialization, we use the pre-trained weights obtained with the Grassland Europe and DeepWeeds datasets since they are semantically closest to the Rumex segmentation task. The whole Rumex Denmark 2020 dataset is considered since it is already relatively small with 282 images.

weight init	pre-trained dataset	mIoU
Kaiming	-	0.758
ImageNet	ImageNet	0.836
SwAV	Grassland Europe	0.830
ImageNet → SwAV	Grassland Europe	0.834
SwAV	DeepWeeds	0.820
ImageNet → SwAV	DeepWeeds	0.839

B.6 Conclusion

In this work, we investigated the potential of self-supervised pre-training in the context of agricultural images. As example, we applied one of the recent popular contrastive learning methods—SwAV [Car+20]—to three different agricultural datasets. For each dataset, we generated data-specific pre-trained weights and show their clear success for classification downstream tasks, in particular improving label efficiency in semi-supervised settings. Furthermore, the computational cost for generating data-specific pre-trained weights is fairly low, allowing one to generate easily new pre-trained weights for any custom model architecture. We also showed that SwAV is less effective for segmentation downstream tasks, showing no significant improvement on the mIoU performance as shown in Tables B.6 and B.7. SwAV belongs to the group of instance discrimination methods, learning features on the image level but not explicitly on the pixel level, which can be considered crucial for segmentation downstream tasks. With the late success of contrastive instance discrimination methods, it is expected that contrastive self-supervised approaches on the pixel level will gain attention within research in computer vision to increase performance on more complex downstream tasks. Just recently, work in that area has been published [Xie+21]. In our experiments, we clearly showed the potential of self-supervised approaches in the domain of agriculture, and therefore, future developments will be followed carefully and taken into consideration for segmentation downstream tasks.

B.7 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258.

References

[Arg+20] David Argüeso et al. "Few-Shot Learning approach for plant disease classification using images taken in the field." In: *Computers and Electronics in Agriculture* 175 (2020), page 105542. ISSN: 0168-1699.

[Azi+21] Shekoofeh Azizi et al. "Big Self-Supervised Models Advance Medical Image Classification." In: *ArXiv* abs/2101.05224 (2021).

[Bos+20] Petra Bosilj et al. "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture." In: *J. Field Robotics* 37.1 (2020), pages 7–19.

[BRB18] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pages 4413–4421.

[BU17] Suchet Bargoti and James Underwood. "Deep fruit detection in orchards." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pages 3626–3633.

[Car+18] Mathilde Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features." In: *European Conference on Computer Vision*. 2018.

[Car+20] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments." In: *Advances in Neural Information Processing Systems*. Edited by H. Larochelle et al. Volume 33. Curran Associates, Inc., 2020, pages 9912–9924.

[CH21] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pages 15750–15758.

[Che+17] Nived Chebrolu et al. "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields." In: *The International Journal of Robotics Research* (2017). DOI: 10.1177/0278364917720510.

[Che+20] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations." In: *Proceedings of the 37th International Conference on Machine Learning*. Edited by Hal Daumé III and Aarti Singh. Volume 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pages 1597–1607.

[Chi+20] Mang Tik Chiu et al. "Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[CMR12] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "Algorithms for Learning Kernels Based on Centered Alignment." In: *J. Mach. Learn. Res.* 13.1 (March 2012), pages 795–828. ISSN: 1532-4435.

[Cri+02] Nello Cristianini et al. "On Kernel-Target Alignment." In: *Advances in Neural Information Processing Systems*. Edited by T. Dietterich, S. Becker, and Z. Ghahramani. Volume 14. MIT Press, 2002.

[Cut13] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport." In: *Advances in Neural Information Processing Systems*. Edited by C. J. C. Burges et al. Volume 26. Curran Associates, Inc., 2013.

[Den+09] Jia Deng et al. “Imagenet: A large-scale hierarchical image database.” In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pages 248–255.

[DGE15] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. “Unsupervised Visual Representation Learning by Context Prediction.” In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pages 1422–1430.

[Dos+14] Alexey Dosovitskiy et al. “Discriminative Unsupervised Feature Learning with Convolutional Neural Networks.” In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’14. Montreal, Canada: MIT Press, 2014, pages 766–774.

[dos+19] Alessandro dos Santos Ferreira et al. “Unsupervised deep learning and semi-automatic data labeling in weed discrimination.” In: *Computers and Electronics in Agriculture* 165 (2019), page 104963. ISSN: 0168-1699.

[Esp+20a] Borja Espejo-Garcia et al. “Improving weeds identification with a repository of agricultural pre-trained deep neural networks.” In: *Computers and Electronics in Agriculture* 175 (2020), page 105593. ISSN: 0168-1699.

[Esp+20b] Borja Espejo-Garcia et al. “Towards weeds identification assistance through transfer learning.” In: *Computers and Electronics in Agriculture* 171 (2020), page 105306. ISSN: 0168-1699.

[GH10] Michael Gutmann and Aapo Hyvärinen. “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.” In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Edited by Yee Whye Teh and Mike Titterington. Volume 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pages 297–304.

[Gri+20] Jean-Bastien Grill et al. “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning.” In: *Advances in Neural Information Processing Systems*. Edited by H. Larochelle et al. Volume 33. Curran Associates, Inc., 2020, pages 21271–21284.

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised Representation Learning by Predicting Image Rotations.” In: *International Conference on Learning Representations*. 2018.

[Gül+21] Ronja Güldenring et al. “Few-leaf Learning: Weed Segmentation in Grasslands.” In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[He+15] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, 2015, pages 1026–1034. ISBN: 9781467383912.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 770–778.

[He+19] Tong He et al. "Bag of Tricks for Image Classification with Convolutional Neural Networks." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pages 558–567.

[He+20] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pages 9726–9735.

[Jac12] Jaccard. "The distribution of the flora of the alpine zone." In: *New Phytologist*. Volume 11. Carg se, France., 1912, pages 37–50.

[JT21] L. Jing and Y. Tian. "Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey." In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 43.11 (November 2021), pages 4037–4058. ISSN: 1939-3539.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Edited by Yoshua Bengio and Yann LeCun. 2015.

[Kor+19] Simon Kornblith et al. "Similarity of Neural Network Representations Revisited." In: *Proceedings of the 36th International Conference on Machine Learning*. Edited by Kamalika Chaudhuri and Ruslan Salakhutdinov. Volume 97. Proceedings of Machine Learning Research. PMLR, September 2019, pages 3519–3529.

[KTN19] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Deep learning-based visual recognition of rumex for robotic precision farming." In: *Computers and Electronics in Agriculture* 165 (2019), page 104973. ISSN: 0168-1699.

[LH17] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Mad+19] Simon L. Madsen et al. "Generating artificial images of plant seedlings using generative adversarial networks." In: *Biosystems Engineering* 187 (2019), pages 147–159. ISSN: 1537-5110.

[Mar+20] Paula A Marin Zapata et al. "Self-supervised feature extraction from image time series in plant phenotyping using triplet networks." In: *Bioinformatics (Oxford, England)* (October 2020). ISSN: 1367-4803.

[MYA17] Mostafa Mehdipour Ghazi, Berrin Yanikoglu, and Erchan Aptoula. "Plant identification using deep neural networks via optimization of transfer learning parameters." In: *Neurocomputing* 235 (2017), pages 228–235. ISSN: 0925-2312.

[MZH16] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. "Unsupervised Learning using Sequential Verification for Action Recognition." In: *CoRR* abs/1603.08561 (2016).

[NF16] Mehdi Noroozi and Paolo Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles." In: *Computer Vision – ECCV 2016*. Edited by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pages 69–84. ISBN: 978-3-319-46466-4.

[Ols+19] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning." In: *Scientific Reports* 9.2058 (1 February 2019). DOI: 10.1038/s41598-018-38343-3.

[OLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding." In: *ArXiv* abs/1807.03748 (2018).

[Pat+16] Deepak Pathak et al. "Context Encoders: Feature Learning by Inpainting." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 2536–2544.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Volume 9351. LNCS. Springer, 2015, pages 234–241.

[Sko+19] Søren Skovsen et al. "The GrassClover Image Dataset for Semantic and Hierarchical Species Understanding in Agriculture." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pages 2676–2684.

[Suh+18] Hyun K. Suh et al. "Transfer learning for the classification of sugar beet and volunteer potato under field conditions." In: *Biosystems Engineering* 174 (2018), pages 50–65. ISSN: 1537-5110.

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Edited by Yoshua Bengio and Yann LeCun. 2015.

[Sze+16] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision." In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[Wan+14] Jiang Wang et al. "Learning Fine-Grained Image Similarity with Deep Ranking." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pages 1386–1393.

[Xie+21] Zhenda Xie et al. "Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning." In: (2021).

[YPB16] Jianwei Yang, Devi Parikh, and Dhruv Batra. "Joint Unsupervised Learning of Deep Representations and Image Clusters." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 5147–5156.

[Zha+20] Zhicheng Zhao et al. "When Self-Supervised Learning Meets Scene Classification: Remote Sensing Scene Classification Based on a Multitask Learning Framework." In: *Remote Sensing* 12.20 (2020). ISSN: 2072-4292.

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization." In: *ECCV*. 2016.

PAPER C

RumexWeeds: A grassland dataset for agricultural robots

Ronja Güldenring[1,*], Frits van Evert[2] and Lazaros Nalpantidis[1]

*Corresponding author: ronjag@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
[2] Agrosystems Research, Wageningen University & Research, PO Box 16, 6700 AA Wageningen, The Netherlands

Abstract

Computer vision can lead towards more sustainable agricultural production by enabling robotic precision agriculture. Vision-equipped robots are being deployed in the fields to take care of crops and control weeds. However, publicly available agricultural datasets containing both image data as well as data from navigational robot sensors are scarce. Our real-world dataset RumexWeeds targets the detection of the grassland weeds: *Rumex obtusifolius L.* and *Rumex crispus L.*. RumexWeeds includes whole image sequences instead of individual static images, which is rare for computer vision image datasets, yet crucial for robotic applications. It allows for more robust object detection, incorporating temporal aspects and considering different viewpoints of the same object. Furthermore, RumexWeeds includes data from additional navigational robot sensors — GNSS, IMU and odometry — which can increase robustness, when additionally fed to detection models. In total the dataset includes 5,510 images with 15,519 manual bounding box annotations collected at 3 different farms and 4 different days in summer and autumn 2021. Additionally, RumexWeeds includes a subset of 340 ground truth pixels-wise annotations. The dataset is publicly available at https://dtu-pas.github.io/RumexWeeds/. In this paper

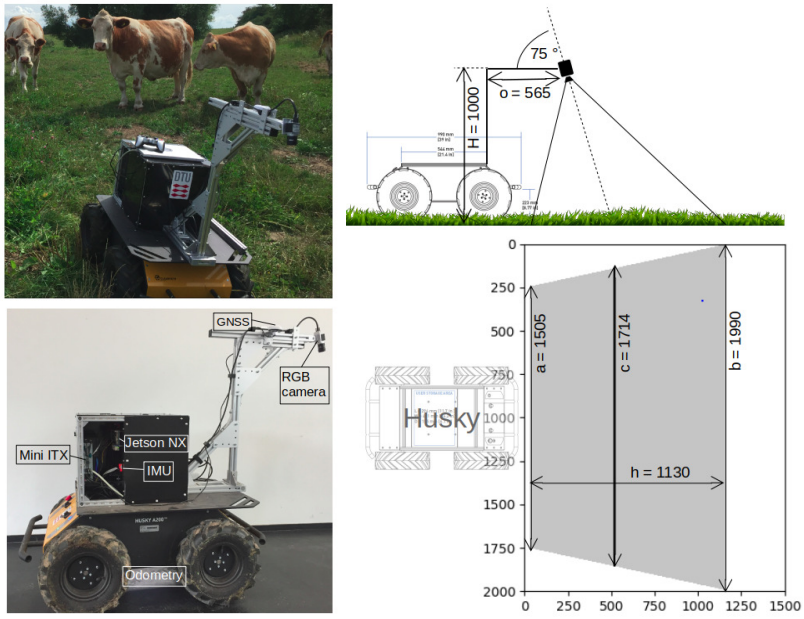
we also use RumexWeeds to provide baseline weed detection results considering a state-of-the-art object detector; in this way we are elucidating interesting characteristics of the dataset.

C.1 Introduction

Weed control is one of the biggest challenges in organic agriculture. Manual weeding is often necessary, but this is expensive, it is physically demanding, and labourers are often hard to find. Robots for precision agriculture have the potential to perform the weed removal task efficiently, paving the way toward more sustainable agriculture. Within dairy farming, *Rumex obtusifolius L.* (broad-leaved dock) and *Rumex crispus L.* (curled dock) are among the most troublesome grassland weeds worldwide because they can grow in a wide range of climatic conditions as well as in many different soil types. As an example, approximately 60 - 70% of grassland fields both in Germany and Japan are heavily infested with both Rumex species [Zal04; Hon86]. Furthermore, it is stated that in Europe, 80% of the herbicides used for weed control in grassland are applied to *Rumex obtusifolius L.* [Zal04]. Rumex is less nutritious than grass for grazing animals, and if consumed in large doses it can get poisonous for the animals, due to the large proportion of nitrate and oxalate in the plant [S H04]. Obviously, Rumex shares soil resources and therefore, competes with sown and other more nutritious pasture species [Zal04]. If Rumex is not controlled, it creates a significant yield loss of 10% - 40% [Eve+11].

The automated detection of *Rumex obtusifolius L.* (broad-leaved dock) has been investigated in agricultural robotics from the early 2000s up until today [Pol+07; Dür+08; SWS08; Eve+09; Ank+10; Ahm+14; KTN16; BCF18; BF17; KTN18; Zha+18b; Val+19; KTN19; SAŠ19; Lam+21; Gül+21; GN21]. This underlines the application's relevance and the need for controlling Rumex efficiently and robustly at dairy farms. The above-mentioned research groups have evaluated their work on their own, relatively small datasets. These datasets have not been made publicly available in most cases.

We aim to close that gap by making a new 30.3 GB large dataset—RumexWeeds—publicly available, enabling comparability of future approaches. RumexWeeds consists of image sequences with a total of 5,510 images of 2.3 MP resolution and 15,519 manual bounding box annotations collected at 3 different farms and on 4 different days in the summer and autumn of 2021. Additionally, navigational robot sensor points from GNSS, IMU and odometry are recorded. The dataset is publicly available at `https://dtu-pas.github.io/RumexWeeds/`https://dtu-pas.github.io/RumexWeeds/. In this paper, apart from introducing the RumexWeeds dataset, we also present detection as well as segmentation prediction results on it, which serve as a baseline for the research community.



(a) Robot in the real world **(b)** Camera System Setup

Figure C.1: The agricultural robot platform. (a) We build upon the basic configuration of the Clearpath Husky robot with an odometry sensor by adding a RGB camera, a low-cost GNSS sensor, an IMU sensor, and a Nvidia Jetson Xavier NX. (b) The RGB camera is pointing towards the ground with an angle of 75° and at a height of $H = 1000\ mm$, resulting in a trapezoidal field coverage. Due to the mounting offset of $o = 565\ mm$ the image plane doesn't include any robot parts.

C.2 Related Work

In the domain of grassland agriculture, only three relevant and publicly available datasets were identified.

The *Rumex Ancenis dataset* [KTN19] contains grassland images with *Rumex obtusifolius L.* bounding box annotations. However, the dataset consists only of images taken within one field and one day as well as under controlled lighting conditions. The variety of the image data is therefore low. The *GrassClover* dataset [Sko+19] targets the segmentation of clover and grass in order to estimate the biomass composition on grassland fields. The images are generated synthetically using image composition, i.e. no extensive manual pixel-wise annotations were required. Therefore, the images are not fully representative of real weeds. The *DeepWeeds* dataset [Ols+19] is a large image classification dataset including eight common Australian weeds in their natural environment.

The generation of a sole image dataset—as the ones above—requires significantly less effort than the generation of a robotics dataset, where not only images but also data from additional robot sensors, such as GPS, IMU, odometry or LIDAR, are collected. A robotics dataset must be collected with an actual robot platform and the data management complexity is notably higher. Obviously, these datasets are richer in information and allow one to relate datapoints from different sensors, resulting in different possible applications, like e.g. visual SLAM and object mapping. In the following, we summarize three relevant robotics datasets within agricultural field robotics—even though, none of them targets grasslands.

The *SugarBeet* dataset [Che+17] was the first agricultural field robotics dataset collected with the *BoniRob* platform [Che+17]. Data was recorded over three months in a sugar beet farm, including a high number of different sensors: RGB camera, stereo camera, laser scanner, GPS, and odometry. Furthermore, it provides pixel-wise ground truth annotation for a sub-selection of 400 images. For the *Brassica* dataset [BWS20], weekly sequences of cauliflower and broccoli were collected to capture the growth cycle. Their robot platform *Ladybird* is equipped with a GPS, IMU, a forward-pointing line-scan hyper-spectral camera, a downwards-pointing stereo camera in controlled lighting conditions as well as a downwards-pointing thermal camera. For a selected subset of the acquired images, ground-truth annotations on pixel– as well as bounding-box–level are provided. Finally, the *Rosario* dataset [Pir+19] is a multi-sensor dataset—stereo camera, IMU, GPS, odometry—targeting specifically the application of visual SLAM. The stereo camera is pointing towards the forward-driving direction and no image annotations are provided.

So far, agricultural robots are predominantly applied to crop cultivation fields, where plants are clearly separable from the soil by means of simple color thresholding or by considering near-infrared (NIR) imagery that has higher reflectivity on chlorophyll content. Wíth RumexWeeds, we provide the first robotics dataset for grasslands. RumexWeeds targets the most problematic weed in dairy farming *Rumex*, which is expected to be more challenging as foreground vegetation (weeds) and background (grass) share similar color and chlorophyll content.

C.3 The Agricultural Robot Platform

The Clearpath Husky robot serves as the base platform, equipped with four relevant sensors (Figure C.1a). These sensors as well as the computing unit is specified in the following. - The camera has been calibrated intrinsically with the ROS camera calibration tool [1]. For the extrinsic calibration, a checkerboard is attached to an aluminum beam, which is temporarily attached to the robot and reaches out to the camera's field coverage. The beam allows one to determine the exact checkerboard position and by detecting the checkerboard marker in the camera image, the exact camera position can finally be determined.

[1] http://wiki.ros.org/camera_calibration

GNSS sensor. We mounted a Navilock NL-8022MU USB 2.0 Multi GNSS receiver on the top of the camera beam in order to receive the best possible signal. The low-cost GNSS sensor is based on the u-blox 8 chipset and has a positioning accuracy of 2.5 m circular error probable (CEP). Therefore, the sensor provides a rough absolute robot position but is not by itself sufficient for precise navigation or localization tasks.

IMU sensor. The Xsens MTI-630 sensor provides $\pm 0.2°$ accuracy for roll/pitch and $\pm 1°$ accuracy for yaw. It is mounted inside of the black cube structure on board the Husky platform (see again Figure C.1a).

Odometry sensor. The odometry is measured with the wheel encoders that come by default with the Husky robot platform: 1024 pulses per revolution (PPR) optical incremental quadrature encoders mounted pre-gearbox, which result in approximately 72000 counts per meter.

Computer Setup. The robot is equipped with two separate computing units: (1) a Mini-ITX Single Board Computing System, which is part of the Husky platform, and (2) an embedded Single Board Computer Nvidia Jetson Xavier NX. Both units communicate via Ethernet. The Mini-ITX is equipped with 2.7 GHz Intel i5-4570TE, 8 GB RAM, 120 GB SATA2 SSD, 2x RS-232, 10x USB 2.0, 2x GigE, PCI Express, 120W power supply, and Wi-Fi adapter. It runs the ROS Master and handles robot-relevant tasks. The navigational sensors, i.e. GPS, IMU, and odometry are connected to the Mini-ITX. The Jetson Xavier NX has a 6-core NVIDIA Carmel Arm processor, a 384-core NVIDIA Volta GPU, and 8 GB of RAM and handles vision-relevant tasks, such as data collection or object detection. The Basler camera is connected directly to the Jetson unit.

C.4 Data Acquisition

On grasslands, a weeding robot is expected to be deployed to the fields approximately 3-4 weeks after the grass has been cut. At this point in time, the Rumex weeds have re-grown to such an extent, that they are clearly visible and standing out from the grass; however, they have not yet reached a plant size, which is difficult to handle autonomously. Therefore, the data acquisition for the RumexWeeds dataset has always been performed at a weed growth of approximately 3-4 weeks. Note, that we can still observe weed size variances within the data because the growing speed is also dependent on the prevailing weather conditions during the growing period, as well as the soil or other conditions.

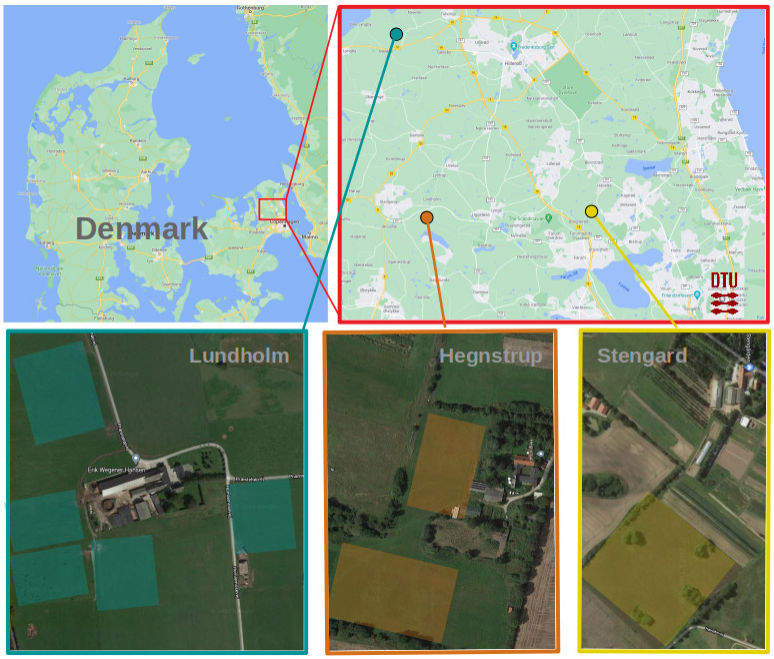


Figure C.2: During data acquisition, three different farms (Lundholm, Hegnstrup, Stengard) in the hinterland of Copenhagen in Denmark were visited in order to capture a variety of grassland compositions. Images were collected approximately 3-4 weeks after the grass has been cut.

Species composition of grassland can vary strongly from field to field, therefore we considered three different dairy farms in the Copenhagen hinterland in Denmark as shown in Figure C.2: (1) Lundholm, (2) Hegnstrup, (3) Stengard. If available, even different fields within the same farm were considered, which are marked in Figure C.2. Furthermore, data was acquired on four different days during the summer and autumn of 2021. The exact dates are specified in Table C.1. We captured data during all relevant weather conditions: clear/sunny, partly cloudy/partly sunny, mostly cloudy, and right after it rained, leading to data acquisition on wet ground.

During data collection, the robot was manually controlled, keeping an average speed of 0.4 m/s. We didn't cover the whole field, but drove the robot criss-cross through the field, focusing on relevant and interesting parts of the field including either Rumex weed plants or other plants that can easily be confused with Rumex (called *negative plants* in the following). The data was collected with the *rosbag*-tool, which saves the sensor data on their publishing frequency, which is 5 FPS for the RGB camera, 20 FPS for the IMU, and 10 FPS for the GNSS and the Odometry.

Table C.1: Summary of annotated datapoints from the 5 different data collection sessions on four different days and three different farms. It includes the number of foreground (FG) images—images including one or more Rumex objects—, the number of pure background (BG) images, and the number of sequences. Furthermore, the number of objects for both classes (*Rumex obtusifolius L.*, *Rumex crispus L.*) and for both annotation types (bounding box, pixel-wise) are listed. Note, that the bounding box annotations are performed for the whole dataset—therefore, additional characteristics namely the total proportion of positive pixels vs all image pixels (below stated as: *pos pixel proportion*), as well as the average bounding box size as a percentage of the whole image size are given (below stated as: *avg. object size*).

	ID	20210806_hegnstrup	20210806_stengard	20211006_stengard	20210807_lundholm	20210908_lundholm
General	Date	Aug. 06, 2021		Oct. 11, 2021	Aug. 7, 2021	Sep. 8, 2021
	Farm Name	Hegnstrup	Stengard		Lundholm	
	Location	55°50'12.5"N 12°22'57.8"E	55°50'24.5"N 12°12'46.8"E		55°56'36.7"N 12°10'09.0"E	
	sequences	18	21	16	29	14
	# FG imgs	454	1459	1207	786	962
	# BG imgs	126	75	61	342	38
Bounding Box Annotation	# Rumex obtusifolius objects	563	3653	4358	774	3208
	# Rumex crispus objects	89	1296	1142	264	172
	pos pixel proportion	7.57 %	5.66 %	6.22 %	4.11 %	12.90 %
	avg. object size	6.74 %	1.75 %	1.44 %	4.47 %	3.82 %
Pixel-wise Annotation	# Rumex obtusifolius objects	23	66	92	17	71
	# Rumex crispus objects	3	16	41	7	4

C.5 Image Annotation

C.5.1 Bounding Box Annotation

For all labeled image sequences, bounding box annotations were generated manually for the grassland weed Rumex. Each bounding box includes the whole plant with all attached leaves. If the plant consists of only one leaf, the single leaf is enclosed by a bounding box. On very dense weed images, it can be difficult to identify all plants with high certainty. Here, noisy labels can be expected.

Furthermore, we differentiate between the two relevant sub-species of Rumex, namely *Rumex obtusifolius L.* and *Rumex crispus L.*, and assign the relevant class to each bounding box. Both sub-species are equally undesired on dairy grassland fields and therefore, for most applications, it is reasonable to treat both species as one class. The assignment decision in one of the above-mentioned classes was made purely based on the visual appearance in the images; no real-world plant phenotype such as root size, flowering branches or seed weight was considered. Since the majority of plants are captured in a relatively small growing state with no flowers present, the only way to visually differentiate between the two sub-species is by considering the leaf and stem characteristics, as described by Cavers and Harper [CH64] and summarized in Table C.2. During the annotation process, we experienced that the leaves' shape was the most important characteristic for our class assignment decisions.

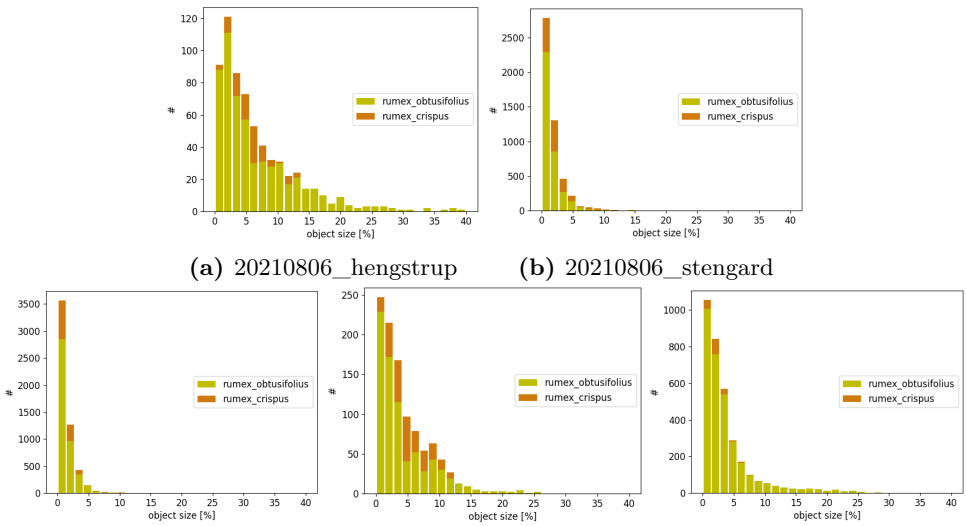
We developed a custom annotation tool in order to take advantage of the image sequences as well as the navigational robot data. Initial bounding box annotations

Table C.2: Weed phenotype of *Rumex obtusifolius L.* vs. *Rumex crispus L.* from Cavers and Harper [CH64]. Only leaf—and stem—characteristics are listed because they are visible in the image data of RumexWeeds.

	Rumex obtusifolius L.	***Rumex crispus L.***
basal leaf length	up to 30 cm	up to 40 cm
basal leaf width	up to 15 cm	up to 11 cm
leaf phenotype	• ovate-oblong • cordate at base • apex rounded • margins undulate	• lanceolate/oblong-laceolate • cauline leaves less crisped than base leaves • leaf margins vary from nearly flat to strongly crisped
stem phenotype	• stem glabrous • stem deeply furrowed with reddish patches	• petiole up to 6 cm length

are tracked using odometry measurements as well as visual CRST [Luk+18]. If both trackers agree, the tracked bounding box is proposed to the annotator. The annotator can accept, modify or delete the proposal.

Table C.1 summarizes the datapoints that have been collected and annotated on four different days and at three different farms. It includes the number of foreground images—the ones including one or more Rumex objects, the number of pure background images and the total number of bounding box annotations for both classes (*Rumex obtusifolius L.*, *Rumex crispus L.*). Since bounding box annotations are performed for the whole dataset, we provide additional characteristics with the total proportion of positive pixels vs all image pixels, as well as the average bounding box size as a percentage of the whole image size. Although we have a significantly larger number of foreground images—i.e. images containing at least one object, the Rumex plant is still highly under-represented when considering the positive pixel proportion for each data collection session. A more detailed distribution of object size as a percentage of the image size is given in Figure C.3. In Figure C.4, the object distribution over the image plane is shown for all labeled foreground images. It can be clearly seen, that weed plants are more frequently present in the center of the image. This can be explained by the fact, that during data acquisition the robot was teleoperated manually with a joystick, aiming to catch as many weed plants as possible. The operators unconsciously tended to drive centered over the weeds.



(a) 20210806_hengstrup **(b)** 20210806_stengard

(c) 20211006_stengard **(d)** 20210807_lundholm **(e)** 20210908_lundholm

Figure C.3: Distribution of the object size as a percentage of the image size for both weed classes (yellow: *Rumex obtusifolius L.*, yellow: *Rumex crispus L.*) and the different data collection sessions.

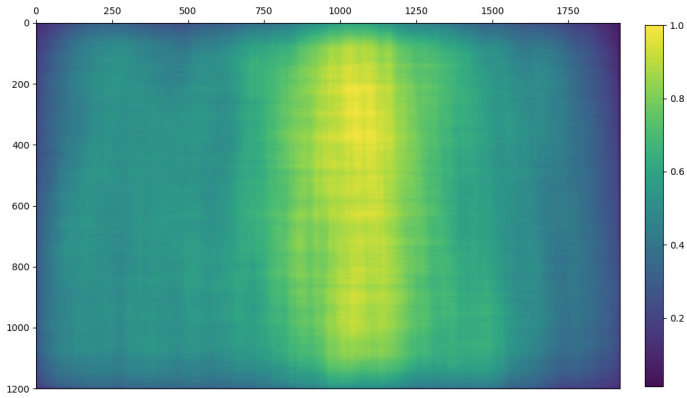


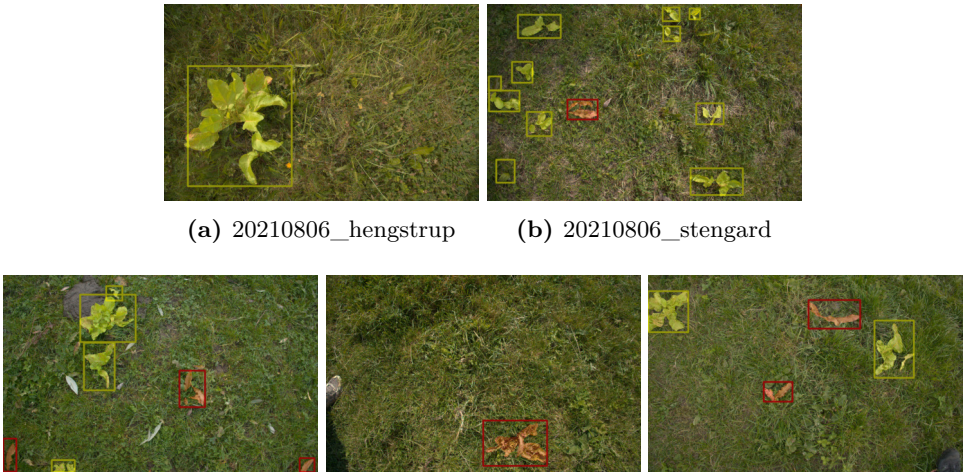
Figure C.4: Object distribution within the image plane for all foreground images. The weed object frequency is significantly higher in the image center due to human bias, while teleoperating the robot.

C.5.2 Pixel-wise Annotation

Additionally, we used CVAT [Sek+20] to provide a small number of carefully manually annotated ground truth masks for a random subset of 20 images per location and day, resulting in 100 images and 340 segmented bounding box crops in total. Figure F.5 shows sample images of RumexWeeds with bounding box annotations and the corresponding manually segmented bounding box crops.

C.6 Dataset Structure

From the data acquisition process, we get raw rosbag-files as stated in Section C.4. In a post-processing step, we synchronize the recorded datapoints and extract relevant fractions in more general data formats. The overall structure of the extracted and clean dataset is illustrated in Figure C.6. RumexWeeds contains 5 subfolders—one for each dataset collection specified in Table C.1. Each dataset collection contains an arbitrary number of sequences, while each sequence consists of a number of consecutive images at 5 FPS. The corresponding datapoints of IMU, GNSS, and Odometry are saved as a dictionary in the `seq<seq_id>/*.json` files. We simply save the corresponding ROS message as a dictionary, which is `sensor_msgs/NavSatFix.msg` for the GNSS, `sensor_msgs/Imu.msg` for the IMU, and `nav_msgs/Odometry.msg` for the Odometry data. The images and the navigational data are linked via the image file name, which serves as a key in the json-dictionary. In other words, all sensor points with the same key are time-synchronized. The file `seq<seq_id>/rgb.json` simply contains the timestamp in the standard timestamp format and is therefore



(a) 20210806_hengstrup **(b)** 20210806_stengard

(c) 20211006_stengard **(d)** 20210807_lundholm **(e)** 20210908_lundholm

Figure C.5: Sample images from *RumexWeeds* with bounding box annotations and corresponding manual pixel-wise annotation of these bounding box crops for the different data collection sessions. A selection of the *RumexWeeds* dataset variety is presented: sunny vs. cloudy, plant size variation, plant density, negative objects such as cowpat, negative plants, tree leaves, and a human foot. Note the color encoding of the two classes, which is yellow for *Rumex obtusifolius L.* and orange for *Rumex Crispus L.*.

redundant with the timestamp information embedded in the image name. The `seq<seq_id>/annotations.xml` contains all ground truth (i.e. performed by a human) bounding box and segmentation annotations for the corresponding image sequence. The format of the annotations follows the conventions of the CVAT annotation tool version 1.1 [Sek+20]. Finally, the `metadata.json` includes information concerning all sequences within one data collection `<collection_id>`. It contains the intrinsic camera coefficients as well as transforms from base_link to the three sensors: → camera_link, → imu_link, → base_gnss as well as the transform base_footprint → base_link. The base_footprint-frame is positioned on the ground, while the base_link-frame lays within the robot, therefore giving insight into the robot's height above ground. The folder `dataset_splits` contains dataset splits in order to compare ones results to our baseline results in Section C.7 and C.8.

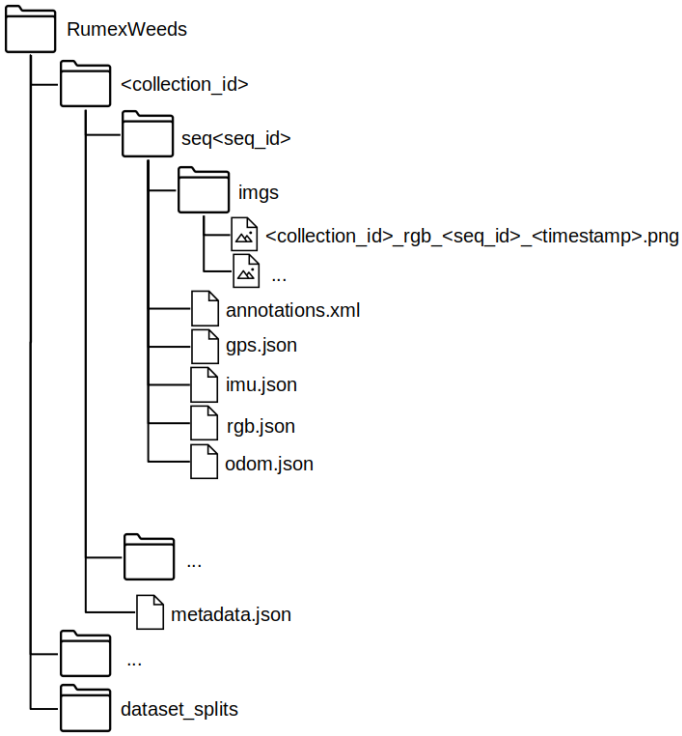


Figure C.6: Folder structure of RumexWeeds, containing 5 subfolders—one for each data collection session—containing an arbitrary number of shorter and longer image sequences. Each image in the sequence is related to additional navigational datapoints from IMU, GNSS and Odometry, saved in `*.json` format. Furthermore, ground-truth annotations for *Rumex obtusifolius L.* and *Rumex crispus L.* are saved in `annotations.xml`.

One might wonder, how the image sequences have been extracted. In order to increase the proportion of positive objects within the data, we focused on extracting sequences containing positive images with Rumex objects. A sequence starts when a Rumex weed enters the image frame and ends when an adequate number of consecutive negative images follow. As a result, we get sequences of different lengths: single Rumex occurrences result in short sequences, while Rumex-dense areas result in longer sequences. We also considered fractions with rare negative objects, including e.g. similar looking negative plants, cow pats or shadows. In Figure C.7, we show the number of images as well as the number of objects for each extracted sequence. It can be assumed, that sequences with a significantly higher number of objects than images indicate a high weed density.

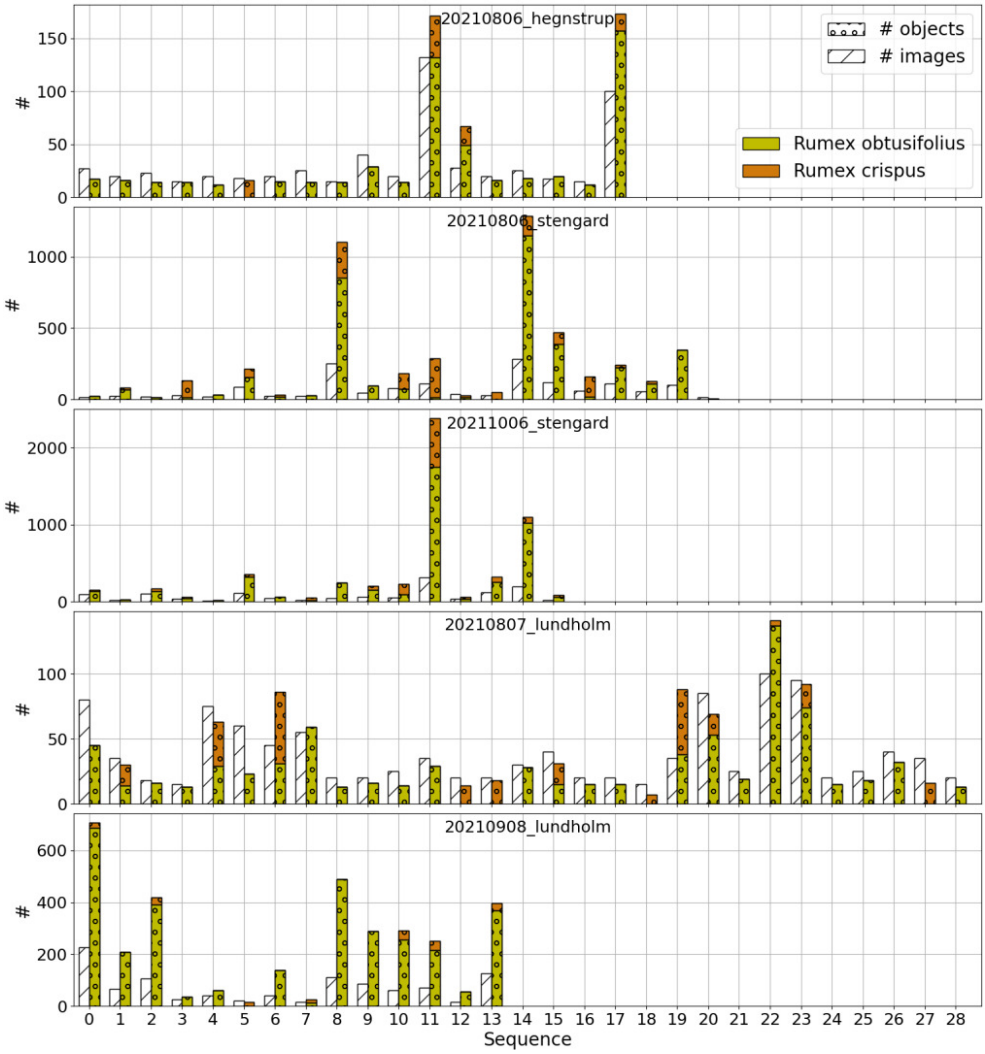


Figure C.7: The number of images as well the number of objects is shown for each sequence within the 5 different data collection sessions. Sequences with a significantly higher number of objects than images indicate a high weed density.

C.7 Rumex Detection Baseline

In this section, we present baseline results on Rumex detection, considering our RumexWeeds dataset. Our targeted application is a Rumex weeding robot, which requires a rough Rumex position in order to successfully approach and remove the

weed. We treat both species *Rumex obtusifolius L.* and *Rumex crispus L.* as one class because they are equally undesired and are removed in the same way by the robot.

C.7.1 Experimental Setup

C.7.1.1 Training Settings

For our baseline, we solely consider the very recent, state-of-the-art YOLOX architecture [Ge+21]. YOLOX builds upon the well-established object detection architecture YOLOv3 and adopts it to newer state-of-the-art object detection advances. The major architectural modifications are listed in the following:

- Decoupled head. Extracted features are further processed by a decoupled head, while one pathway is dedicated to the classification and the other is dedicated to the localization of the object. It greatly improves the converging speed.

- Anchor-free. Bounding box predictions are not anymore made on pre-defined and domain-specific anchor boxes, but the box parameters are predicted directly (top-left corner, width, and height). This design choice affects both the network size as well as the mAP model performance positively.

- Multi positives. While in YOLOv3 only one ground-truth bounding box is considered during loss computation, YOLOX takes into account multiple ground-truth bounding boxes, which fall within a 3×3 center area of the prediction, inspired by Tian et al. [Tia+19].

- SimOTA. A more advanced label assignment strategy SimOTA [Ge+21] is applied, which selects the k-top positive labels based on a defined cost function. All remaining labels are treated as negative. SimOTA improves the mAP model performance clearly.

In our experiments, we use the original YOLOX implementation [2]. The following parameters have been modified and fixed for all performed experiments: The input image size is resized to $[640 \times 640]$, while standard augmentation such as flip, rotation, color jitter, blur, scale, and brightness/contrast jitter as well as image normalization is applied during training. We didn't experience an improvement with more advanced augmentation methods like mosaic (ulatraliytics-YOLOv3 [3]) or mixup [Zha+18a]; therefore, they are disabled. During training, a batch size of 64 is used and the network is updated with a learning rate of 0.001. All remaining parameters are kept as they are in the default YOLOX settings.

[2] https://github.com/Megvii-BaseDetection/YOLOX
[3] https://github.com/ultralytics/yolov3

C.7.1.2 Model Evaluation

In our experiments, we consider two different types of evaluation protocols.

- Random split: We split all image sequences randomly in train_R, val_R, test_R (2742, 1233, 1535 images). Each split contains images from all different data collection sessions, but it is assured that image sequences are not separated. This is crucial because sequences contain the same plants multiple times from different image view angles.

- New Data split: We want to explore how well the model performs on unseen data collection sessions or even unseen fields. The corresponding data collection (i.e. 20210806_hegnstrup, 20210806_stengard, 20211006_stengard, 20210807_lundholm, 20210908_lundholm) or field (i.e. 20210806_hegnstrup, 20210806_stengard + 20211006_stengard, 20210807_lundholm + 20210908_lundholm) serve as test set, while the remaining data is split randomly in train and validation set as described previously.

The detection performance is measured with the mean Average Precision (mAP) according to the COCO eval toolkit [4], while we report two variants: $mAP_{50:95}$ and mAP_{50}. For each class, the area under the precision-recall curve for predictions exceeding a specific IoU threshold IoU_{thresh} is computed. Finally, the average over all classes is taken. mAP_{50} represents a mAP at $IoU_{thresh} = 0.5$, while $mAP_{50:95}$ additionally averages the mAP over 10 IoU thresholds from 0.5 to 0.95 with an interval of 0.05. In $mAP_{50:95}$ predictions of more precise overlap (i.e. higher IoU) have a stronger impact on the final score.

The mAP is a well-established benchmark metric to compare model performances to each other. However, the complex nature of mAP makes it difficult to interpret specific model characteristics. Depending on the targeted real-world application, some detection errors are less serious than others. For example, tumor detection aims to minimize the number of missing detections. On the other hand, incorrect detections as well as localization errors do not matter so drastically, because they can be corrected manually by an expert when checking the discovery. On the contrary, for our Rumex detection application, it is desired to reduce the number of incorrect detections, because removing other, potentially good plants wastes resources and prolongs the working time of the robot on the field. Missing detections are not so problematic, as long as the majority of weeds is removed successfully. It can be assumed, that the missed weed will be removed in another session when Rumex features have developed further, i.e. the plant grew bigger.

In order to get more in-depth information about the model performance characteristics, we additionally analyze the sources of error according to Bolya et al. [Bol+20]. Hereby, the following errors are possible for our 1-class detection problem:

[4] https://cocodataset.org/#detection-eval

- Localization Error (Loc): The prediction can be assigned to a ground-truth bounding box, but the IoU falls below the IoU_{thresh}.
- Duplicate Detection Error (Dupl): The prediction can be assigned to a ground-truth bounding box with an $IoU \geq IoU_{thresh}$, but there is already another higher-scoring prediction that was assigned to the same ground-truth bounding box.
- Background Error (Bkg): A prediction that does not overlap with any ground-truth bounding box.
- Missed Ground Truth Error (Miss): Undetected ground-truth bounding box, that does not fall under localization error.

Bolya et al. [Bol+20] determine the above-mentioned errors individually by correcting one error at a time and re-computing the mAP_{50}. The difference between the corrected mAP_{50} and the original mAP_{50} can be traced back to the corresponding error.

C.7.2 Experimental Results

C.7.2.1 Impact of model size

Table C.3 lists the detection performance as well as the inference speed for different YOLOX variants, which differ in their model size. Generally, deeper models tend to achieve better detection results, to the disadvantage of increased inference time. For RumexWeeds, we can not observe a significant detection boost for larger models like *YOLOX-m* and *YOLOX-s*. Since RumexWeeds targets real-time deployment on autonomous vehicles, we will consider *YOLOX-tiny* in the following experiments. *YOLOX-tiny* achieves a reasonable inference speed of 21.4 ms per frame on the Jetson NX (i.e. 46.7 FPS), while keeping the detection performance relatively close to the deeper models. Our inference speed is reported for FP16-precision and a batch size of $bs = 1$. On the Jetson, we additionally converted the model to tensorRT [NVI21].

Table C.3: Detection performance as well as inference speed for the different YOLOX variants, which differ in their model size, i.e. number of model parameters and flops.

					Average Inference Time [ms]	
model name	Params (M)	Flops (G)	$mAP_{50:95}$	mAP_{50}	Tesla V100	Jetson Xavier NX
YOLOX-m	25.3	73.5	27.8	54.1	14.7	48.7
YOLOX-s	8.9	26.6	27.1	51.4	12.2	25.9
YOLOX-tiny	5.0	15.1	26.9	52.8	11.1	21.4
YOLOX-nano	0.9	2.5	26.4	52.2	13.88	18.9

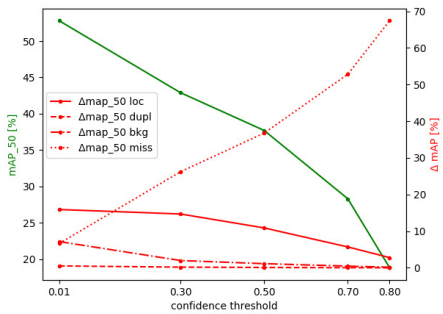
C.7.2.2 Impact of detection confidence score

Each object prediction comes with a confidence score, which represents the network's confidence in the correctness of the corresponding bounding box prediction. For real-world applications, the confidence threshold is of great importance: Only predictions above a certain confidence threshold are considered in order to provide reliable predictions with a low number of False Positive predictions (FP). However, benchmarked detection models use rather low, unrealistic confidence thresholds, e.g. the default confidence threshold of YOLOX is set to 0.01.

For our Rumex detection application, it is most important to reduce the number of Background predictions to minimize the resources during the weeding process. In return, some missed or less accurately located predictions can be accepted. In Table C.4, the performance of our *YOLOX-tiny* model from the previous experiment is listed for different confidence thresholds. Hereby, we report the overall $mAP_{50:95}/mAP_{50}$ as well as the different sources of error: Localization Error (Loc), Duplicate Detection Error (Dupl), Background Error (Bkg) and Missed Ground Truth Error (Miss). The results confirm that with increasing confidence threshold the overall mAP reduces. On the other hand, predictions are more reliable, because the share of Background and Localization error decreases significantly, while the share of Missed Ground Truth Error increases.

Table C.4: YOLOX-tiny is evaluated for different confidence thresholds. The overall mAP decreases with increasing confidence threshold, but remaining predictions get more reliable: We can observe a decrease of Localization and Background Error at the cost of an increase of missing predictions. The accompanying Figure visualizes the results from this table.

			ΔmAP_{50}			
conf thresh	$mAP_{50:95}$	mAP_{50}	Loc	Dupl	Bkg	Miss
0.01	26.9	52.8	15.9	0.5	7.2	6.4
0.3	22.9	42.9	14.7	0.2	2.0	26.2
0.5	20.7	37.7	10.9	0.1	1.1	36.8
0.7	16.8	28.3	5.7	0.0	0.5	52.8
0.8	12.3	18.8	2.8	0.0	0.2	67.4



C.7.2.3 Impact of different image views

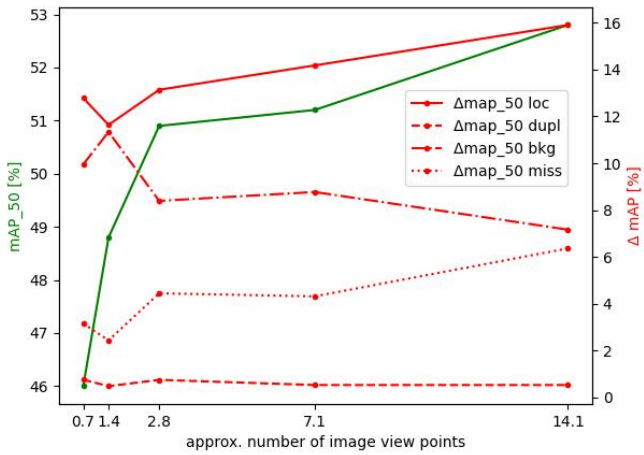
As discussed, RumexWeeds includes whole image sequences instead of single images of the targeted object, which is particularly rare for image datasets. The availability of image sequences has two main advantages:

- It allows one to take into consideration temporal image data, profiting from historical information. Since Rumex weeds tend to appear in clusters on the field, this is extremely valuable information. The detection of a plant such as Rumex should become more likely with the increasing density of Rumex weeds. We will not investigate this further within this paper, because it would go beyond its scope.
- Each object will appear several times from different viewpoints throughout the image plane. Since our camera is tilted by 75°, it is expected that plants within the lower half of the image provide better viewpoints with more relevant features. However, it is desired to detect the weed as soon as possible to ensure forward-looking navigation.

Considering the vertical field coverage of $h = 1130mm$, an average robot speed of $0.4\frac{m}{s}$, and an image acquisition frame rate of 5 FPS, we can assume that a plant occurs approximately from $(1.13/0.4 * 5) = 14.1$ different viewpoints. In Table C.5, we experimentally show the impact of multiple viewpoints. A *YOLOX-tiny* model is trained on the random training set of reduced FPS, leading to fewer viewpoints per plant. Finally, the model is evaluated on the original random validation set with the original number of approximately 14.1 viewpoints per plant. The object detection performance as well as the accompanying error analysis for reduced image acquisition frame rates of $[2.5, 1, 0.5, 0.25]$ FPS – corresponding to an approximate number of viewpoints per plant of $[0.7, 1.4, 2.8, 7.1, 14.1]$ – is shown in Table C.5. For the increasing number of viewpoints, we can observe two relevant effects: (1) The overall $mAP_{50:95}/mAP_{0.5}$ improves; (2) The share of strongly undesired background predictions decreases, while the share of missing prediction increases. From this, it can be concluded, that a distinction between Rumex weeds and negative plants of similar appearance gets less robust with decreasing number of viewpoints.

Table C.5: YOLOX-tiny is trained on a different number of viewpoints per plant. With an increasing number of viewpoints, the overall mAP performance as well as the share of undesired background predictions improves. The accompanying Figure visualizes the results from this table.

approx. num view points	$mAP_{50:95}$	mAP_{50}	ΔmAP_{50}			
			Loc	Dupl	Bkg	Miss
0.7	22.6	46.0	12.8	0.8	10.0	3.2
1.4	23.7	48.8	11.6	0.5	11.3	2.4
2.8	25.4	50.9	13.2	0.7	8.4	4.4
7.1	26.2	51.2	14.2	0.5	8.8	4.3
14.1	26.9	52.8	15.9	0.5	7.2	6.4

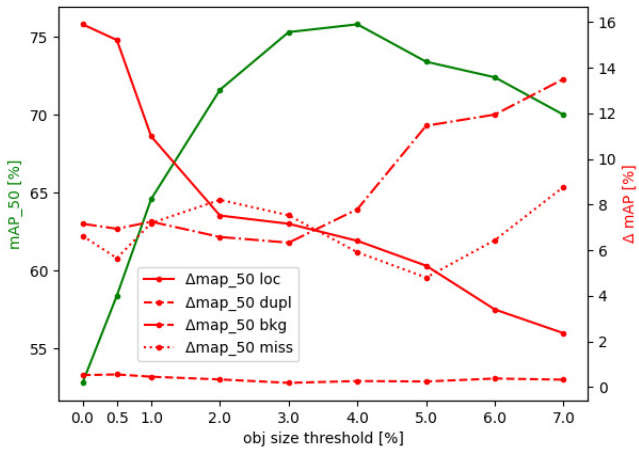


C.7.2.4 Impact of object size

It is generally known that object detection of small objects is less robust, because of the lack of relevant features. Furthermore, for RumexWeeds, the label noise of small Rumex annotations is expected to be higher, because even for a human it is challenging to distinguish small Rumex plants from small negative plants. In Table C.6, we evaluate our YOLOX-tiny model for different object sizes. During evaluation, only ground truth annotations as well as predictions that are above the *object size threshold* are considered. Again, the *object size threshold* is listed as a percentage of the input image size. The results in Table C.6 approve the expectation, that the detection performance increases with increasing *object size threshold*. However, on larger *object size threshold* $> 4\%$, the performance decreases slightly. A possible explanation for this decrease is, that the dataset contains a relatively low amount of very large object sizes with an overall average object size and standard deviation of $2.48\% \pm 3.37\%$.

Table C.6: YOLOX-tiny performance is evaluated for different object size thresholds as a percentage of the input image. The detection performance increases significantly when discarding smaller objects during evaluation. However, for larger object size thresholds $> 4\%$, the performance starts dropping slightly. The accompanying Figure visualizes the results from this table.

obj size thresh [%]	$mAP_{50:95}$	mAP_{50}	ΔmAP_{50}			
			Loc	Dupl	Bkg	Miss
0	26.9	52.8	15.9	0.5	7.2	6.4
1	35.5	64.6	11.0	0.5	7.3	7.2
3	48.6	75.3	7.2	0.2	6.3	7.5
4	49.8	75.8	6.4	0.3	7.8	5.9
5	47.6	73.4	5.3	0.2	11.5	4.8
7	45.7	70.0	2.4	0.3	13.5	8.8



C.7.2.5 Model performance on unseen days and farms

Table C.7 includes the results for our day– and farm–specific evaluation protocol to show how well the model generalizes over unseen data. Again, *YOLOX-tiny* is trained on the corresponding training–set. We take the best model checkpoint according to the validation–set to finally assess the model performance on the test–set, which includes unseen data from either a different data collection session, i.e. new day, or even from a completely new farm.

In the following, we give a possible explanation for the performance increase/decrease for the different experiments in Table C.7, compared to the performance on the random split in experiment 0.

- Experiment 1: Decreased performance, because one specific negative plant looks very similar to *Rumex* and occurs very frequently in 20210807_lundholm, but not in the other data collection sessions. The high share of Background Error supports this statement.
- Experiment 2 and 7: Increased performance, because the average object size is relatively high for the data collection session in the test set. Furthermore, no exceptionally large amount of similar-looking negative plants are present in the image data.
- Experiment 3: The performance on farm Lundholm is affected by the characteristics of the two data collections of experiments 1 and 2. The performance lies somewhat in between both individual experiments.
- Experiment 4, 5, 6: Overall decreased detection performance, because the average object size is significantly lower for these data collection sessions/farm. The model does not learn well to detect small objects from the training data.

Table C.7: YOLOX-tiny is evaluated for unseen data collection sessions (i.e. days) as well for unseen farms. Depending on the different data collection session, performance increases as well as decreases can be observed, which represents the difficulty of the different data collection session. The difficulty increases if the data includes a high share of small objects or new negative plants, that were not present in the other data collection session.

Exp.	Set				ΔmAP_{50}			
	train/val	test	$mAP_{50:95}$	mAP_{50}	loc	dupl	bkg	miss
0	train_R val_R	test_R	29.0	55.9	14.9	0.5	6.9	6.8
1	20210806_hegnstrup 20210806_stengard 20211006_stengard 20210908_lundholm	20210807_lundholm	26.2	46.6	6.1	0.2	23.5	1.6
2	20210806_hegnstrup 20210806_stengard 20211006_stengard 20210807_lundholm	20210908_lundholm	37.3	63.4	10.0	0.5	8.4	6.7
3	20210806_hegnstrup 20210806_stengard 20211006_stengard	20210807_lundholm 20210908_lundholm	31.7	54.2	8.5	0.2	19.3	2.1
4	20210806_hegnstrup 20210908_lundholm 20211006_stengard 20210807_lundholm	20210806_stengard	22.3	46.6	16.7	0.6	8.1	6.2
5	20210806_hegnstrup 20210908_lundholm 20210806_stengard 20210807_lundholm	20211006_stengard	18.6	40.9	21.6	0.6	6.3	6.5
6	20210806_hegnstrup 20210908_lundholm 20210807_lundholm	20210806_stengard 20211006_stengard	16.4	34.9	18.9	0.5	5.1	13.2
7	20210806_stengard 20211006_stengard 20210908_lundholm 20210807_lundholm	20210806_hegnstrup	37.5	62.3	7.8	0.1	14.2	1.3

C.8 Pixel-wise box crop segmentation baseline

Bounding Box annotations provide a rough position of the weed plant in the image. For some applications, weed segmentation is crucial in order to estimate a more precise plant size and position. Unfortunately, particularly manual pixel-wise object annotations are expensive to obtain for large amounts of training data. Therefore, only a very small amount of pixel-wise ground truth masks has been created for RumexWeeds. We hypothesize, that these few ground truth segmentation masks are sufficient to generate rough masks for bounding box detections. Performing segmentation predictions for bounding box crops simplifies the problem to a great extent, because we only need to differentiate between foreground and background, while the foreground mask inherits the label of the bounding box. Furthermore, it is generally expected that no other negative plant objects with similar features (e.g. Taraxacum, Cirsium) lay within the bounding box crops.

C.8.1 Experimental Setup

C.8.1.1 Training Settings

As Network architecture, we use the popular and solid CNN segmentation architecture U-Net [RPB15] with an Encoder-Decoder structure. During training we keep some settings fixed: We apply transfer-learning to the model encoder by loading pre-trained ImageNet [Den+09] weights. The network is updated using the Adam Optimizer [KB15] with a learning rate of $lr_{enc} = lr_{dec} = 0.001$ as well as the Lovasz Loss [BB17]. Furthermore, we apply cosine learning rate scheduling [LH17]. Standard image augmentations such as random flipping, random rotation, and random scaling are applied.

C.8.1.2 Model Evaluation

Our pixel-wise manually annotated bounding box crops are split into training– and test–set (proportion 8:2), while the test–set will only be considered during the assessment of the final model. Since we have limited data, we will use k-fold-cross-validation (with k=5) on the training–set to select the best model among all. As performance measurement, we consider the *mean Intersection over Union (mIoU)* – also called *Jaccard Index* [Jac12]. For background as well as foreground class, a separate IoU is computed by dividing the number of intersecting pixels between prediction and ground truth with the sum of all pixels presented in both. Finally, the mIoU is the averaged sum over both classes.

C.8.2 Experimental Results

Table C.8 documents how the U-Net model performance has been incrementally improved by tuning some selected hyper-parameters. Increasing the image size as well

as using a deeper model architecture (i.e. deeper encoder) had a significant impact on the model performance. Finally, we used the best model settings (ID 4) and increased the number of training epochs until saturation occurred. On the test–set, the final model (ID 7) achieves a slightly better mIoU score of 0.776 compared to the training mIoU of 0.770.

Table C.8: Tuning of the convolutional segmentation network U-Net [RPB15]. We can observe strong impacts on the model performance for increased input image resolution and the depth of the network. Finally, the model of ID 4 performs best and we increase the number of training epochs until the performance saturates (ID 6, 7).

ID	Encoder			Input size W/H	No. epochs E	mean IoU
	Resnet-18	ResNet-34	VGG16			
1	✓			128	20	0.702
2	✓			256	20	0.737
3	✓			512	20	0.746
4		✓		512	20	0.759
5			✓	512	20	0.740
6		✓		512	40	0.764
7		✓		512	80	**0.770**

In Figure C.8, we show some qualitative results of the test–set with ground truth masks in the top row and model predictions in the bottom row. The majority of segmentation masks look very satisfying and are definitely precise enough in order to determine an approximate plant size as well as root center. As soon the plant leaves get relatively thin and resemble grass structure, the predictions are less reliable, which mainly occurs for the specie *Rumex crispus L.*.

(a) Ground truth masks

(b) Rumex Crop Predictions

Figure C.8: Visual comparison of ground truth masks (a) to the predictions of model 7 (b) of the test–set. The majority of predictions look satisfying. Lower quality predictions can be observed when the plant leaves are very thin and resemble grass structures. Note the color encoding of the two classes, which is yellow for *Rumex obtusifolius L.* and orange for *Rumex Crispus L.*.

C.9 Conclusion

In this work, we introduced a new robotics dataset, RumexWeeds, which addresses the detection of the problematic weeds *Rumex obtusifolius L.* as well as *Rumex crispus L.* in grasslands. The dataset includes whole annotated image sequences as well as additional navigational sensor data from IMU, GNSS, and Odometry, collected with the Clearpath Husky robot platform. As the first publicly available agricultural dataset of its kind, we hope that the structure of the RumexWeeds dataset will allow and motivate the robotics community to develop more advanced detection algorithms that go beyond standard computer vision approaches, such as YOLO [Red+16] or SSD [Liu+16] that take individual images as input. Temporal image information as well as the additional navigational data can be considered in the detection model to potentially increase the detection robustness and efficiency.

Making the first step of actively using RumexWeeds for weed detection, we also present detection as well as segmentation prediction results using state-of-the-art approaches. Our results not only can serve as a baseline for the research community, but also elucidate interesting characteristics of the dataset and the considered weed detection task.

C.10 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258. The authors would like to thank the organic farmers Erik Hansen (Lundholm), Allan Clausen (Hegnstrup), and Otto Stengaard (Stengaard) for offering their fields for data collection and insightful discussions regarding Rumex treatment, as well as Peter Langborg Wejse from Arla Foods for the support. The authors would like to thank Bastian Munch Kjøller for his extraordinary effort in designing and building up a solid sensor mount on our Husky robot.

References

[Ahm+14] Faisal Ahmed et al. "Automated weed classification with local pattern-based texture descriptors." In: *Int. Arab J. Inf. Technol.* 11 (2014), pages 87–94.

[Ank+10] Thomas Anken et al. "Automatic detection of broad-leaved dock in grassland." In: January 2010.

[BB17] Maxim Berman and Matthew B. Blaschko. "Optimization of the Jaccard index for image segmentation with the Lovász hinge." In: *CoRR* abs/1705.08790 (2017).

[BCF18] A. Binch, N. Cooke, and Charles W Fox. "Rumex and Urtica detection in grassland by UAV." In: *14th International Conference on Precision Agriculture*. 2018.

[BF17] A. Binch and C.W. Fox. "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland." In: *Computers and Electronics in Agriculture* 140 (August 2017), pages 123–138.

[Bol+20] Daniel Bolya et al. "TIDE: A General Toolbox for Identifying Object Detection Errors." In: *ECCV*. 2020.

[BWS20] Asher Bender, Brett Whelan, and Salah Sukkarieh. "A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica." In: *Journal of Field Robotics* 37.1 (2020), pages 73–96.

[CH64] P.B. Cavers and J.L Harper. "Rumex obtusifolius L. and R. crispus L." In: *Journal of Ecology* 52.3 (1964), pages 747–766.

[Che+17] Nived Chebrolu et al. "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields." In: *The International Journal of Robotics Research* (2017).

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pages 248–255.

[Dür+08] Lorenz Dürr et al. “Machine vision detection and microwave-based elimination of rumex obtusifolius L. on grassland.” In: 2008.

[Eve+09] F K van Evert et al. “Real-time vision-based detection of Rumex obtusifolius in grassland.” In: *Weed Research* 49.2 (2009), pages 164–174.

[Eve+11] Frits K. van Evert et al. “A robot to detect and control broad-leaved dock (Rumex obtusifolius L.) in grassland.” In: *Journal of Field Robotics* 28.2 (2011), pages 264–277.

[Ge+21] Zheng Ge et al. “YOLOX: Exceeding YOLO Series in 2021.” In: *arXiv preprint arXiv:2107.08430* (2021).

[GN21] Ronja Güldenring and Lazaros Nalpantidis. “Self-supervised contrastive learning on agricultural images.” In: *Computers and Electronics in Agriculture* 191 (2021), page 106510. ISSN: 0168-1699.

[Gül+21] Ronja Güldenring et al. “Few-leaf Learning: Weed Segmentation in Grasslands.” In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[Hon86] Akio Hongo. “Infestation of Rumex obtusifolius L. and Distribution Pattern of Its Individual Plants in Sown Grasslands in Eastern Hokkaido.” In: *Journal of Weed Science and Technology* 31.4 (1986), pages 300–305.

[Jac12] Jaccard. “The distribution of the flora of the alpine zone.” In: *New Phytologist*. Volume 11. Carg se, France., 1912, pages 37–50.

[KB15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” In: *CoRR* abs/1412.6980 (2015).

[KTN16] T. Kounalakis, G. Triantafyllidis, and L. Nalpantidis. “Weed recognition framework for robotic precision farming.” In: *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*. 2016, pages 466–471.

[KTN18] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. “Image-Based Recognition Framework for Robotic Weed Control Systems.” In: *Multimedia Tools Appl.* 77.8 (April 2018), pages 9567–9594. ISSN: 1380-7501.

[KTN19] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. “Deep learning-based visual recognition of rumex for robotic precision farming.” In: *Computers and Electronics in Agriculture* 165 (2019), page 104973. ISSN: 0168-1699.

[Lam+21] Olee Hoi Ying Lam et al. “An open source workflow for weed mapping in native grassland using unmanned aerial vehicle: using Rumex obtusifolius as a case study.” In: *European Journal of Remote Sensing* 54.sup1 (2021), pages 71–88.

[LH17] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Liu+16] Wei Liu et al. "SSD: Single Shot MultiBox Detector." In: *ECCV (1)*. Edited by Bastian Leibe et al. Volume 9905. Lecture Notes in Computer Science. Springer, 2016, pages 21–37. ISBN: 978-3-319-46447-3.

[Luk+18] Alan Lukežič et al. "Discriminative Correlation Filter with Channel and Spatial Reliability." In: *International Journal of Computer Vision* 126 (July 2018). DOI: 10.1007/s11263-017-1061-3.

[NVI21] NVIDIA. *TensorRT*. 2021.

[Ols+19] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning." In: *Scientific Reports* 9.2058 (1 February 2019). DOI: 10.1038/s41598-018-38343-3.

[Pir+19] Taihú Pire et al. "The Rosario dataset: Multisensor data for localization and mapping in agricultural environments." In: *The International Journal of Robotics Research* 38.6 (2019), pages 633–641.

[Pol+07] G. Polder et al. "Weed Detection Using Textural Image Analysis." English. In: *EFITA/WCCA Conference*. 2007.

[Red+16] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pages 779–788.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Volume 9351. LNCS. Springer, 2015, pages 234–241.

[S H04] P. Doležal S. Hejduk. "Nutritive value of broad-leaved dock (Rumex obtusifolius L.) and its effect on the quality of grass silages." In: *Czech Journal of Animal Science* 49 (2004), pages 144–150.

[SAŠ19] Damian Schori, Thomas Anken, and Dejan Šeatović. "Using Fully Convolutional Networks for Rumex Obtusifolius Segmentation, a Preliminary Report." In: *2019 International Symposium ELMAR*. 2019, pages 119–122.

[Sek+20] Boris Sekachev et al. *opencv/cvat: v1.1.0*. Version v1.1.0. August 2020. DOI: 10.5281/zenodo.4009388.

[Sko+19] Søren Skovsen et al. "The GrassClover Image Dataset for Semantic and Hierarchical Species Understanding in Agriculture." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pages 2676–2684.

[SWS08] Dejan Seatović, Zhaw Winterthur, and Switzerland. "3D-Object Recognition, Localization and Treatment of Rumex Obtusifolius in its Natural Environment." In: *International Conference on Precision Agriculture*. June 2008.

[Tia+19] Zhi Tian et al. "FCOS: Fully Convolutional One-Stage Object Detection." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pages 9626–9635.

[Val+19] J. Valente et al. "DETECTING RUMEX OBTUSIFOLIUS WEED PLANTS IN GRASSLANDS FROM UAV RGB IMAGERY USING DEEP LEARNING." In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W5 (2019), pages 179–185.

[Zal04] J G Zaller. "Ecology and non-chemical control of Rumex crispus and R. obtusifolius (Polygonaceae): a review." In: *Weed Research* 44 (2004), pages 414–432.

[Zha+18a] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization." In: *International Conference on Learning Representations*. 2018.

[Zha+18b] W. Zhang et al. "Broad-Leaf Weed Detection in Pasture." In: *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. 2018, pages 101–105.

PAPER D

Real-time joint-stem prediction for agricultural robots in grasslands using multi-task learning

Jiahao Li[1], Ronja Güldenring[1] and Lazaros Nalpantidis[*1]

[*]Corresponding author: lanalpla@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

Abstract

Autonomous weeding robots need to accurately detect the joint stem of grassland weeds in order to control those weeds in an effective and energy-efficient manner. In this work, keypoints on joint stems and bounding boxes around weeds in grasslands are detected jointly using Multi-Task Learning. We compare a two-stage, heatmap-based architecture to a single-stage, regression-based architecture—both based on the popular YOLOv5 object detector. Our results show, that introducing joint stem detection as a second task boosts the individual weed detection performance in both architectures. Furthermore, the single-stage architecture clearly outperforms its competitors with an OKS of 56.3 in joint stem detection, while also achieving real-time performance of 12.2 FPS on Nvidia Jetson NX, suitable for agricultural robots. Finally, we make the newly created joint stem ground truth annotations publicly available at https://dtu-pas.github.io/RumexWeeds/ for the relevant research community.

D.1 Introduction

This paper addresses the perception need for robotic weeding in grasslands, thus contributing to precision farming. While weed detection is a problem often addressed in the relevant literature, the state of the art rarely considers the exact detection of the detected weeds' joint stems in grasslands. Nevertheless, the precise localization of the joint stem is crucial if robots are to autonomously control weeds on grasslands, e.g. by means of laser weeding, electrocution, or other targeted control methods.

Toward this direction, we have investigated potential multi-task learning (MTL) approaches that optimize jointly two relevant tasks—keypoint detection for joint stems as well as bounding box detection for grassland weeds (plants)—leading to improved performance. We apply two different architectures, namely YOLO-HRNet and YOLO-Pose, both of which are based on the popular YOLOv5 object detector. We use YOLO-HRNet, a heatmap-based architecture operating in two stages, as our baseline approach, whereas we show that YOLO-Pose, which operates in a single stage and is regression-based, can achieve impressive results in terms of detection precision and inference speed for both considered tasks. Even though the considered network architectures are used unmodified, it is not trivial to deploy a full-scale working deep learning approach in environments as challenging as grasslands. Thus, the novelty of this work lies in addressing, for the first time, joint-stem prediction of weeds in grasslands.

Figure D.1: The "GALIRUMI" EU project robot operating on grasslands needs to detect weeds (yellow bounding boxes) and their stem joints (cyan keypoints) to control them in an effective and efficient manner, through precisely targeted laser weeding or electrocution.

Our approach is a significant step toward robotic precision farming. Robotic manipulation requires precise localization of the manipulation target, and agricultural robotics is no exception. Targeted, organic-farming-friendly weed-control approaches,

such as laser weeding and electrocution, require the precise location of the joint stem to target the laser beam or position the electrode to maximize the damaging effect of the energy used. Therefore, fine-grained analysis of the detected weeds is needed to reveal the ideal intervention target position, as shown in Figure D.1 for the "GALIRUMI" EU project robot.

The contribution of this paper is threefold: (i) First, we showcase that MTL is a viable approach to boost performance in the weed detection task in grasslands by integrating joint stem detection as a second task, and furthermore, (ii) we experimentally show that YOLO-Pose not only achieves superior results in both tasks but also achieves real-time detection speeds suitable for agricultural robots. Last but most importantly, (iii) to support all the above and contribute to the relevant research community, we make the newly created joint stem ground truth annotation publicly available as an extension of our RumexWeeds dataset at `https://dtu-pas.github.io/RumexWeeds/` (accessed: 11.9.2023).

D.2 Related Work

Weed detection has been widely investigated in the past decade and it has been posed as a classification problem [Far+19; Smi+19; Dad+20; Wu+20; Ree+22; Gar+22], detection problem via bounding boxes [Yu+19; Jia+20; Jin+22; Zha+22; GEN23], as well as segmentation problem [Faw+19; Cha+20; Gül+21; Sod+22; FSA23]. The object detection architecture YOLO has evolved greatly over the last decade resulting in different popular variants, such as YOLOv5 [Gle22], YOLOX [Ge+21], and YOLOv8 [JCQ23], which have been widely applied to the agricultural domain [Car+23; WH21; Soz+22; GEN23; Yan+23].

However, agricultural robots that actively interact with detected plants need more fine-grained plant information in order to perform tasks such as weeding, picking, or cutting. This is the reason for an increased interest in joint-stem detection in recent years. Langer et al. [Lan+18] take as input the detected plants and apply geometrical computer vision methods in order to retrieve the joint stems. The parameters of their methods are tweaked to address very specific plants at specific growing stages; therefore, they do not adapt to different cases. Deep neural networks are applied to detect the weeds and crops as well as their joint stem keypoints in a multi-task setting. Lottes et al. [Lot+18] apply an encoder–decoder network to segment plants and predict joint stem positions. On top of a shared encoder, two decoders follow—one for each task. The same authors improve their model robustness by incorporating spatial–temporal information (i.e. consecutive image sequences) to the network [Lot+20]. Unfortunately, Lottes et al. [Lot+18; Lot+20] do not segment the individual plants as instances, which risks multiple treatments of the same plant. In contrast, Weyler et al. [Wey+21] use CenterNet [Dua+19] to simultaneously detect crops/weeds with bounding boxes, plant-center keypoints, and leaf keypoints on an instance basis. Lac et al. [Zha+20] propose a two-stage method, where they use YOLOv4 for fast stem detection, followed by a temporal aggregation algorithm to re-

fine the stem predictions. Note that they predict joint-stem positions with bounding boxes instead of keypoints. Zhang et al. [Zha+20] propose, similarly to [Lot+18], an architecture with one shared backbone but two separate decoder paths to generate segmentation masks for segmenting weeds as well as keypoint predictions for joint stems based on heatmaps. They additionally provide the feature maps from the weed segmentation path to the joint stem prediction path and claim to increase accuracy by doing that. Note that the architecture only handles input images containing no more than a single plant each.

Our work performs simultaneous weed and joint-stem detection in grassland fields, which is an underexplored field in the domain of precision farming. To the best of our knowledge, the publications mentioned above have only been applied to crop fields where plants can be clearly separated from their background (soil). However, grassland environments are more challenging since weeds on grassland are much more difficult to differentiate and joint stems are generally covered by adjacent weeds or grasses.

D.3 Method

In this work, we consider two different architectures, both based on YOLOv5[1] .

1. **YOLO-HRNet** serves as baseline architectures, which is inspired by Mask R-CNN [He+17] while being comparable to YOLOv5-based architectures. Like in Mask R-CNN, the object detection model is followed by a simple segmentation model. We choose HRNet16 [Wan+21] as the second stage model because its architecture is particularly good for keypoint prediction. Because of its two stages and the relatively large model size, it is expected that accurate prediction results are produced at the cost of slower inference speeds.

2. **YOLO-Pose** [Maj+22] proposes an extension of anchor-based architectures to predict keypoints. Compared to YOLO-HRNet, the model size of YOLO-Pose [Maj+22] is reduced and predictions are made within one stage. A significant improvement in inference speed but worse prediction accuracy are expected.

YOLO-HRNet as well as YOLO-Pose can be categorized as multi-task learning because hard-parameter sharing is performed between the two tasks: weed detection (i.e. bounding box detection) and joint-stem detection (i.e. keypoint detection). While both architectures perform weed detection with the same approach, the strategy of predicting joint stems differs between YOLO-HRNet and YOLO-Pose: heatmap-based and two-stage vs. regression-based and one-stage.

[1] https://github.com/ultralytics/yolov5

D.3.1 YOLO-HRNet

Figure D.2 gives an overview of the YOLO-HRNet architecture, where HRNet16 [Wan+21] is integrated as a second stage to YOLOv5. The corresponding feature maps of the predicted bounding boxes serve as input for HRNet, meaning that each input contains only a single object. *ROI align* —proposed by the authors of Mask R-CNN [He+17]—is used to map the feature maps to the expected input size. HRNet outputs a heatmap and the maximum value is chosen as the final position for the keypoint. The forwarded input to the HRNet differs between the training and inference pass of the network. In order to increase the amount of training data for HRNet, feature maps from all scales and all positive detections are forwarded to HRNet. On the contrary, during inference, only the final predictions after non-maximum suppression and of the corresponding scale are forwarded to HRNet.

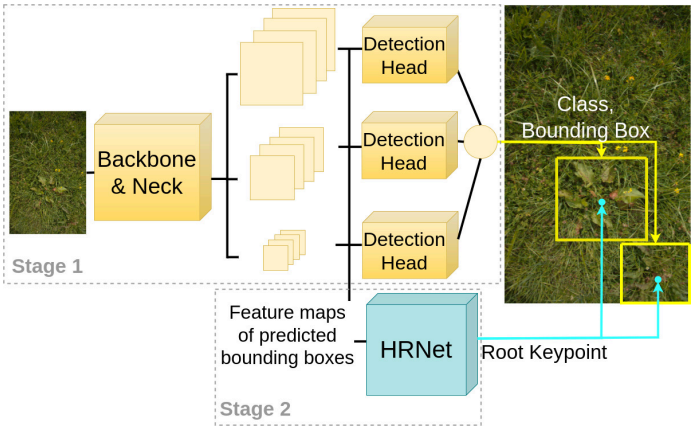


Figure D.2: YOLO-HRNet: HRNet (blue) is integrated in YOLOv5 (yellow) as second stage. The corresponding feature maps of the predicted bounding boxes serve as input to the HRNet and one joint stem keypoint is predicted per input.

The original loss function of YOLO takes into consideration a weighted sum of the classification loss $\mathcal{L}_{class}$, the localization loss $\mathcal{L}_{box}$, and the objectness loss $\mathcal{L}_{obj}$. The objectness loss $\mathcal{L}_{obj}$ computes the binary cross entropy (BCE) of whether a box actually contains an object or not. It can also be understood as a confidence score. The objectness label is set to 1 for boxes that include an object and 0 for boxes without an object. The classification loss $\mathcal{L}_{class}$ computes the BCE for all positive boxes considering the assigned class. Finally, the localization loss $\mathcal{L}_{box}$ uses the Complete-IoU loss (CIoU) as regression criteria for the regressed box coordinates.

The loss function of YOLOv5 is extended with a mean squared error (MSE) loss term, $\mathcal{L}_{MSE}$, handling the keypoint prediction during training. The loss for each scale s and anchor k at location (i, j) is summed up as follows. Note that $w_{box}, w_{obj}, w_{cls}$, and w_{kp} are the corresponding weighting factors in order to weigh

the different loss terms.

$$\mathcal{L}_{YOLO-HRNet} = \sum_{s,i,j,k} w_{box}\mathcal{L}_{box} + w_{obj}\mathcal{L}_{obj} + w_{cls}\mathcal{L}_{cls} + w_{kp}\mathcal{L}_{MSE} \tag{D.1}$$

D.3.2 YOLO-Pose [Maj+22]

Just recently, in 2022, YOLO-Pose [Maj+22] introduced an extension for anchor-based object detection approaches to predict human poses end-to-end. The heatmap-free approach is simple, yet effective: each detection head is extended with a pose head, which regresses keypoints with reference to the anchor center. Figure D.3 gives an overview of the YOLO-Pose architecture.

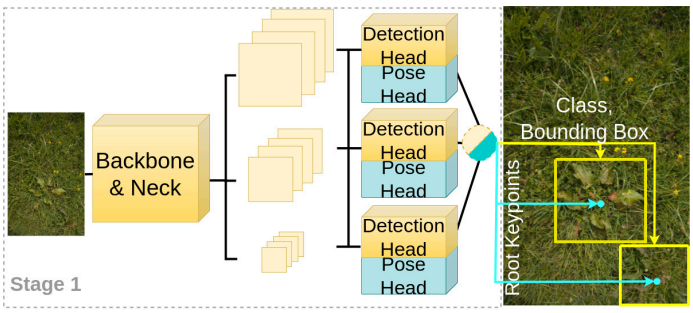


Figure D.3: YOLO-Pose [Maj+22]: YOLOv5 (yellow) is extended with additional *Pose Heads* for each feature scale (blue). The *Pose Heads* regress keypoint positions relative to the anchor box center.

YOLO-Pose introduces a loss function $\mathcal{L}_{OKS}$, that optimizes directly towards the popular metric *object keypoint similarity* (OKS). The loss can only be applied for regression-based methods but not for heatmap-based (i.e. probability maps) approaches such as YOLO-HRNet. For each visible ($v_n > 0$) keypoint n, the exponential is computed for the Euclidean distance between prediction and ground truth d_n^2, divided by the product of the squared scale of the object s^2 and the squared keypoint specific weight $\sigma_{t,n}^2$.

$$\mathcal{L}_{OKS} = 1 - \frac{\sum_n [exp(\frac{d_n^2}{2s^2\sigma_{t,n}^2}) \cdot \delta(v_n > 0)]}{\sum_i [\delta(v_n > 0)]} \tag{D.2}$$

The original YOLOv5 loss function as described previously is extended with $\mathcal{L}_{OKS}$

and is summed up for each scale s and anchor k at location (i, j) as followed.

$$\mathcal{L}_{YOLO-Pose} = \sum_{s,i,j,k} w_{box}\mathcal{L}_{box} + w_{obj}\mathcal{L}_{obj} + w_{cls}\mathcal{L}_{cls} + w_{kp}\mathcal{L}_{OKS} \tag{D.3}$$

D.4 Experimental Setup

D.4.1 Dataset

The publicly available grassland weed dataset RumexWeeds [GEN23] is considered in this work. It is a real-world dataset, targeting the most problematic grassland weed, *Rumex*. Data have been collected with a Husky robot platform, shown in Figure D.1, and its mounted sensors. The dataset includes 98 different image sequences as well as accompanying navigational data from IMU, GPS, and wheel encoders. For all 5510 images, ground-truth bounding box annotations are available, while it is differentiated between two species: *Rumex obtusifolius* and *Rumex crispus*. Please refer to [GEN23] for a more detailed description of the RumexWeeds dataset and some baseline results.

We supplement the data of that dataset with additional manually created keypoint annotations: For each bounding box in the dataset, a joint stem annotation has been performed. Note that even for the human annotator, it can be challenging to identify the precise joint stem position. Therefore, the annotator drew a circular region representing the potential joint-stem position. The joint stem annotation consists of a position (k_{ix}, k_{iy}) as well as a radius r_i, indicating the uncertainty of the human annotator. The higher r_i, the more uncertain the annotator was about the performed annotation. In Figure D.4, a sample image with bounding box as well as joint stem annotations is shown. Our newly created joint stem annotations will be merged and made publicly available at `https://dtu-pas.github.io/RumexWeeds/` (accessed: 11.9.2023).

For our experiments in Section D.5, the data is split in train/val/test with the following ratio 0.70/0.15/0.15. Hereby, we make sure that sequences are not separated between the splits to ensure that the same plant does not appear in different splits, which would falsify the results. Furthermore, like the authors from [GEN23], we treat both Rumex species as one class because they are equally undesired and should both be removed by the agricultural robot.

D.4.2 Metrics

D.4.2.1 Mean Average Precision (mAP)

The object detection task is evaluated with the well-established metric mean average precision (mAP) according to the COCO eval toolkit (`https://cocodataset.org/#detection-eval`, accessed: 11.9.2023), while two variants are reported in our

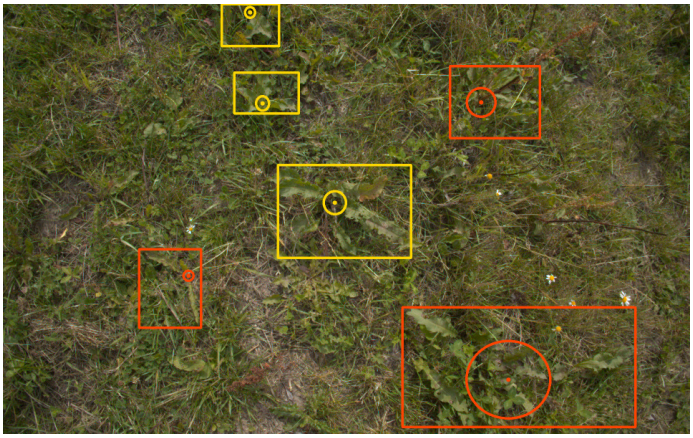


Figure D.4: Example image from the RumexWeeds dataset [GEN23] with the ground-truth bounding boxes and our additional joint stem annotation: Each joint stem has been annotated by the position (k_{ix}, k_{iy}) and a radius r_i, which indicates the uncertainty of the human annotator. Note that it is differentiated between two species: *Rumex obtusifolius* (yellow) and *Rumex crispus* (red).

experiments: $mAP_{50:95}$ and mAP_{50}. The mAP_{50} considers all predictions with an Intersection of Union (IoU) over 0.5 equally as positive examples. On the contrary, $mAP_{50:95}$ averages the mAP over 10 IoU thresholds from 0.5 to 0.95 with an interval of 0.05, which introduces higher weights for higher IoUs. Therefore, $mAP_{50:95}$ also reflects the precision of the predicted bounding box.

D.4.2.2 Object Keypoint Similarity (OKS)

The joint stem detection task is evaluated with object keypoint similarity (OKS), which is the most popular metric for the evaluation of keypoints. As shown in Equation (E.1), for OKS, the Euclidean distance d_i between predicted and ground truth points is taken and normalized by the scale s of the object to eliminate the effect of various object sizes. It is additionally divided by the standard deviation σ between human annotations and true labels, i.e., keypoints that are difficult to identify are weighted with a higher σ. In our experiments, OKS_{50} is utilized to report the joint-stem detection performance.

$$OKS = exp(-\frac{d_i^2}{2s^2\sigma_i^2}) \tag{D.4}$$

D.4.3 Fixed Training Settings

We fix a number of hyperparameters for all our experiments. The input size is (640×640) and during training, the following augmentations are applied to the input images: color jitter, random flip, random scale, and random shift. The networks are updated using the Adam optimizer [KB15]. YOLO-Pose is updated with an initial learning rate of $5e$-4, decaying down to $2e$-1 using cosine learning rate schedule. For YOLO-HRNet, two different initial learning rates are set: $5e$-4 for YOLOv5 and $1e$-3 for HRNet. Again, we use the cosine learning rate schedule to decay both down to $1e$-4. For all models, we have a warm-up phase of 3 epochs. Unless mentioned otherwise, we train for 100 epochs and set the σ in the OKS metric as well as loss to 0.2. All models are trained and evaluated on Nvidia Tesla V100, unless otherwise stated.

D.5 Experimental Evaluation

D.5.1 Ablation Study: Model Complexity

For the computer vision domain, RumexWeeds is a relatively small dataset with 5510 images, and due to its nature of providing image sequences, the variance between consecutive images is reduced. In this ablation study, we determine for both methods—YOLO-HRNet and YOLO-Pose—the model complexity, which leads to the best results without overfitting the data. Each experiment is performed once and the results for both architectures can be found in Table D.1.

For YOLO-HRNet, we vary the size of the shared backbone as well as the input size of the feature map crops to the HRNet in the second stage. As the results show in Table D.1, a bigger input size to HRNet does not lead to significant improvements. Therefore, we fix it to 16, because the smaller the input, the higher the training as well as the inference speed. Increasing the backbone size to DarkNet-53 X leads to slight improvements.

For YOLO-Pose, we vary the size of the shared backbone as well as the complexity of the pose head. Results are presented in Table D.1 and we can conclude that using a more complex pose head leads to improvement in both tasks because we have an increase in mAP as well as OKS. However, when using the biggest backbone model Darknet-53 X, overfitting occurs since both the mAP and OKS reduces compared to Darknet-53 M.

According to our ablation study, we fix the backbone to DarkNet-53 M because it generates very good results for both architectures (YOLO-Pose and YOLO-HRNet) while being relatively small. Furthermore, we use an input size of 16 for the HRNet and a complex head for YOLO-Pose unless mentioned otherwise.

Table D.1: Themodel complexity for both architectures is explored. For both architectures, the model complexity mainly depends on the chosen backbone, while for YOLO-HRNet in (a), the HRNet input size is varied, and for YOLO-Pose in (b), the complexity of the pose head is varied.

Backbone	HRNet Input Size	mAP_{50}	$mAP_{50:95}$	OKS
DarkNet-53 M	16	45.4	23.1	44.1
DarkNet-53 M	24	45.1	**23.8**	44.0
DarkNet-53 M	32	**45.7**	23.3	**44.2**
DarkNet-53 S	16	40.8	20.8	37.0
DarkNet-53 X	16	46.1	23.8	44.2

(**a**) YOLO-HRNet

Backbone	Pose Head	mAP_{50}	$mAP_{50:95}$	OKS
DarkNet-53 M	Simple	52.1	26.3	47.4
DarkNet-53 M	Complex	**54.2**	**27.5**	**50.5**
DarkNet-53 S	Complex	49.1	24.5	48.9
DarkNet-53 X	Complex	52.9	26.3	50.1

(**b**) YOLO-Pose

D.5.2 Ablation Study: Task/Loss Weighting

For multi-task learning, loss weighting is crucial in order to achieve good results. We treat the loss weights as hyperparameters and tune w_{box}, w_{obj}, and w_{kp} using random search. Note, that w_{class} has no relevance in our setup, since we only have one class. In this section, we show the most relevant results.

For YOLO-HRNet (see Table D.2), lowering the w_{kp} relatively to the other weights leads to a performance boost for both tasks, as shown in the second row. On the contrary, further lowering of w_{kp} in the third row increases the mAP only marginally while reducing the OKS drastically. For the following experiments, we fix the weights to $w_{box} = 0.05$, $w_{obj} = 0.85$ and $w_{kp} = 0.10$.

For YOLO-Pose (see Table D.2), lowering w_{kp} relatively to the other weights leads to an improved performance of weed detection while sacrificing a bit of performance in joint-stem detection. However, weed detection is generally favored over joint-stem detection. We fix the weights to $w_{box} = 0.05$, $w_{obj} = 0.95$ and $w_{kp} = 0.01$. In accordance with our previous results, using YOLO-Pose with a complex head boosts the results even further, as shown in the last row of Table D.2.

Table D.2: Hyperparameter tuning of loss weights using random search for YOLO-HRNet (a) and YOLO-Pose (b). We only show the relevant results.

	w_{box}	w_{obj}	w_{kp}	mAP_{50}	$mAP_{50:95}$	OKS
	0.05	0.65	0.30	43.4	22.2	43.2
	0.05	0.85	0.10	45.4	**23.1**	**44.1**
	0.05	0.90	0.05	**46.3**	22.9	37.9

(**a**) YOLO-HRNet

Pose Head	w_{box}	w_{obj}	w_kp	mAP_{50}	$mAP_{50:95}$	OKS
	0.05	0.65	0.15	49.1	22.4	**54.6**
	0.05	0.75	0.1	50.8	25.2	**54.6**
Simple	0.05	0.85	0.05	51.5	25.3	53.7
	0.05	0.95	0.01	**52.1**	**26.3**	47.4
	0.05	0.95	0.005	50.5	25.8	37.8
Complex	0.05	0.95	0.01	**54.2**	**27.5**	**50.5**

(**b**) YOLO-Pose

D.5.3 Ablation Study: Influence of σ_t

In accordance to the metric OKS, the loss function $\mathcal{L}_{OKS}$ weighs each keypoint with σ_t, which represents the standard deviation between human annotation and true labels. However, the true σ is unknown; therefore, we treat it as a hyperparameter and explore its influence on the final prediction. Table D.3 shows that the selection of $sigma_t$ has a significant influence on the prediction performance. When choosing a $sigma_t$ that is too small, i.e., $sigma_t = 0.1$, the introduced annotation noise has a negative impact on the detection performance. With increasing $sigma_t$, label noise is reduced; hence, overfitting can be avoided. However, we can observe that at $\sigma_t > 0.4$, we can see a drastic negative impact on the detection performance. We set $\sigma_t = 0.2$ in our final models.

Table D.3: Influence of different keypoint weighting $sigma_t$ in the loss function $\mathcal{L}_{OKS}$. A low $sigma_t$ introduces too much label noise, leading to reduced detection performance. On the contrary, too high values of $\sigma_t > 0.4$ impact the detection performance negatively likewise. For our final models, we fix $\sigma_t = 0.2$.

σ_t	mAP_{50}	$mAP_{50:95}$	OKS
0.1	50.0	25.6	50.0
0.2	52.2	**26.9**	**50.4**
0.4	**55.0**	26.4	48.1
0.6	52.3	26.9	37.9
0.8	52.7	26.6	34.6

D.5.4 Final Results

In this section, the final models of YOLO-HRNet and YOLO-Pose, as well as a basic YOLOv5 model, are evaluated on the test split. All models are trained once for 200 epochs and the final results are reported on the test split.

Table D.4 shows the weed and joint-stem detection performance for all models. YOLO-Pose is the winner among its competitors while retaining a reasonable training time. The results disprove our expectation: the one-stage architecture (i.e. YOLO-Pose) can keep up with the performance of the two-stage architecture (i.e. YOLO-HRNet) in weed detection and significantly outperforms it in joint-stem detection. Secondly, we can conclude that introducing the joint stem detection task enhances the performance of the weed detection task. The effect can be seen for both YOLO-HRNet and YOLO-Pose. In Figure D.5, we show qualitative results of the final models.

Table D.4: Evaluation of final models on the test split. YOLO-Pose clearly outperforms its competitors. Furthermore, introducing joint-stem detection as a second task enhances the performance in weed detection.

Model	Training Time [h]	mAP_{50}	$mAP_{50:95}$	OKS
YOLOv5	**5.3**	46.5	23.7	/
YOLO-HRNet	42.1	50.0	**27.9**	46.3
YOLO-Pose	5.8	**50.1**	26.1	**56.3**

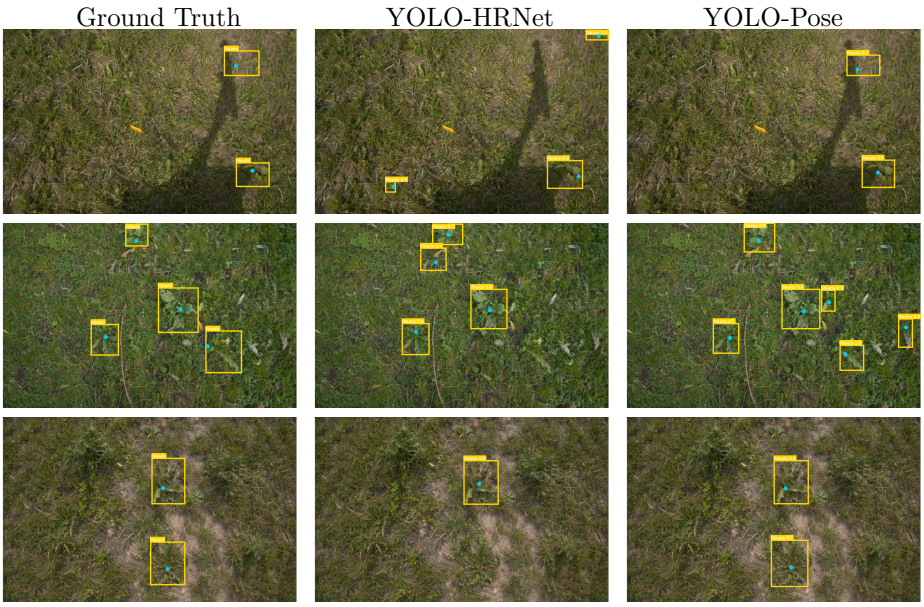


Figure D.5: Qualitative performance of the final models YOLO-HRNet (**middle**), YOLO-Pose (**right**), compared to the ground truth (**left**). Input images are shown with yellow bounding boxes for detected weeds and cyan joint stem positions.

In Table D.5, we present the inference performance. First, we test on a single Nvidia V100 (NVIDIA, Santa Clara, CA, USA) for all final models. As expected, YOLO-HRNet provides a fairly low inference speed due to its two-stage nature. YOLO-Pose is only slightly slower then YOLOv5, which is expected since the YOLO-Pose architecture has additional pose heads. However, its speed is still on the higher end with 101.0/88.5 FPS (simple/complex head) and is a serious candidate for real-time deployment in agricultural robots. Therefore, we deploy YOLO-Pose on the Single Board Computer Nvidia Jetson Xavier NX (NVIDIA, Santa Clara, CA, USA) and get 13.8 and 12.2 FPS for YOLO-Pose with simple and complex heads, respectively, as shown in the lower part of Table D.5.

D.6 Discussion

In our work, we used almost out-of-the-box implementations of popular network architectures, which have the great advantage that relevant pre-trained weights are generally available in order to train efficiently and achieve better results while having limited data available. On the contrary, using a customized architecture requires the generation of pre-trained weights with benchmark datasets, which is costly. In

Table D.5: Inference speed of our final models. YOLO-Pose is slightly slower than YOLOv5 but still suitable for deployment on edge devices onboard agricultural robots.

GPU	Model	# Params	Preproc + NMS [ms]	Inference [ms]	FPS
Nvidia V100	YOLOv5	20.9 M	1.2	8.5	103.1
	YOLO-HRNet	26.6 M		33.6	28.7
	YOLO-Pose (Simple)	20.9 M		8.7	101.0
	YOLO-Pose (Complex)	23.3 M		10.1	88.5
Nvidia Jetson NX	YOLO-Pose (Simple)	20.9 M	5.5	67.0	13.8
	YOLO-Pose (Complex)	23.3 M		76.5	12.2

applied domains, it is common to use those popular architectures with powerful pre-trained weights; e.g., Weyler et al. [Wey+21] used CenterNet [Dua+19] and Lac et al. [Zha+20] is based on Yolov4. More precisely, we combined YOLO with HRNet as a two-stage network and YOLO-Pose as a single-stage network—both of them based on the popular YOLOv5 object detector. It is reasonable to expect that more advanced architectures, such as, e.g., the recent YOLOv8, or proper modifications of them, will result in further gains in terms of performance. However, what we want to showcase is the applicability of such methods on the targeted task, rather than their absolute performance. The scarcity of previous works directly related to our problem reveals that it is not trivial to deploy a full-scale working deep learning approach in environments as challenging as grasslands.

We show that MTL of detecting grassland weeds and their joint stems leads to improved detection performance; therefore, both tasks are complementing each other well. These findings are in accordance with the work of Lotte et al. [Lot+18; Lot+20], who report performance improvements for MTL of plant segmentation and joint-stem detection in crop fields, i.e., on the BoniRob dataset [Che+17]. Our work required the appropriate extension of our existing RumexWeeds dataset. We have extended it with additional joint stem annotations, where each joint stem has been annotated by the position and a radius indicating the uncertainty of the human annotator. As we find the significance of such datasets paramount for advancing the field, we make our new annotations publicly available as part of our RumexWeeds dataset.

It is worth mentioning that before the considered robot can intervene to control the weeds, the 3D positions of the detected keypoints are needed. We take the joint-stem detection and align it with the corresponding depth map in order to retrieve the 3D information. Again, it allows us to use out-of-the-box architectures since the corresponding pre-trained weights are predominantly retrieved from large RGB datasets, such as ImageNet.

D.7 Conclusions

This work has investigated two different multi-task learning approaches to solve jointly the joint-stem and weed detection problems in image data from grasslands. More precisely, we compared a two-stage, heatmap-based architecture (YOLO-HRNet) to a single-stage, regression-based one (YOLO-Pose). Our results show that introducing joint-stem detection as a second task boosts the individual weed detection performance in both architectures. However, YOLO-Pose outperforms its competitors in joint-stem detection with OKS of 56.3 while also achieving real-time performance of 12.2 FPS on an Nvidia Jetson NX device. These characteristics yield our final YOLO-Pose model as an ideal candidate for implementation on autonomous weeding robots that need precise localization of weeds' joint stems to control them.

D.8 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258. The authors would like to thank the organic farmers Erik Hansen (Lundholm), Allan Clausen (Hegnstrup), and Otto Stengaard (Stengaard) for offering their fields for data collection and insightful discussions regarding Rumex treatment.

References

[Car+23] Angelo Cardellicchio et al. "Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors." In: *Computers and Electronics in Agriculture* 207 (2023), page 107757. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2023.107757.

[Cha+20] Julien Champ et al. "Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots." In: *Applications in Plant Sciences* 8.7 (2020), e11373. DOI: https://doi-org.proxy.findit.cvt.dk/10.1002/aps3.11373.

[Che+17] Nived Chebrolu et al. "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields." In: *The International Journal of Robotics Research* (2017). DOI: 10.1177/0278364917720510.

[Dad+20] Mojtaba Dadashzadeh et al. "Weed Classification for Site-Specific Weed Management Using an Automated Stereo Computer-Vision Machine-Learning System in Rice Fields." In: *Plants* 9.5 (2020). ISSN: 2223-7747. DOI: 10.3390/plants9050559.

[Dua+19] Kaiwen Duan et al. "CenterNet: Keypoint Triplets for Object Detection." In: October 2019, pages 6568–6577. DOI: 10.1109/ICCV.2019.00667.

[Far+19] Adnan Farooq et al. "Multi-Resolution Weed Classification via Convolutional Neural Network and Superpixel Based Local Binary Pattern Using Remote Sensing Images." In: *Remote Sensing* 11.14 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11141692.

[Faw+19] Mulham Fawakherji et al. "Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation." In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 2019, pages 146–152. DOI: 10.1109/IRC.2019.00029.

[FSA23] Hafez Fathipoor, Reza Shah-hosseini, and Hossein Arefi. "CROP AND WEED SEGMENTATION ON GROUND-BASED IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORK." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4/W1-2022 (January 2023), pages 195–200. DOI: 10.5194/isprs-annals-X-4-W1-2022-195-2023.

[Gar+22] Francisco Garibaldi-Márquez et al. "Weed Classification from Natural Corn Field-Multi-Plant Images Based on Shallow and Deep Learning." In: *Sensors* 22.8 (2022). ISSN: 1424-8220. DOI: 10.3390/s22083021.

[Ge+21] Zheng Ge et al. "YOLOX: Exceeding YOLO Series in 2021." In: *CoRR* abs/2107.08430 (2021).

[GEN23] Ronja Güldenring, Frits K. van Evert, and Lazaros Nalpantidis. "RumexWeeds: A grassland dataset for agricultural robotics." In: *Journal of Field Robotics* n/a.n/a (2023). DOI: https://doi.org/10.1002/rob.22196.

[Gle22] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. November 2022. DOI: 10.5281/zenodo.7347926.

[Gül+21] Ronja Güldenring et al. "Few-leaf learning: Weed segmentation in grasslands." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pages 3248–3254.

[He+17] Kaiming He et al. "Mask R-CNN." In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pages 2980–2988. DOI: 10.1109/ICCV.2017.322.

[JCQ23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *YOLO by Ultralytics*. Version 8.0.0. January 2023.

[Jia+20] Honghua Jiang et al. "CNN feature based graph convolutional network for weed and crop recognition in smart farming." In: *Computers and Electronics in Agriculture* 174 (2020), page 105450. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105450.

[Jin+22] Xiaojun Jin et al. "A novel deep learning-based method for detection of weeds in vegetables." In: *Pest Management Science* 78.5 (2022), pages 1861–1869. DOI: https://doi.org/10.1002/ps.6804.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *CoRR* abs/1412.6980 (2015).

[Lan+18] Ferdinand Langer et al. "Geometrical Stem Detection from Image Data for Precision Agriculture." In: *CoRR* abs/1812.05415 (2018).

[Lot+18] Philipp Lottes et al. "Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming." In: June 2018. DOI: 10.1109/IROS.2018.8593678.

[Lot+20] Philipp Lottes et al. "Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming." In: *Journal of Field Robotics* 37.1 (January 2020), pages 20–34. ISSN: 1556-4959. DOI: 10.1002/rob.21901.

[Maj+22] Debapriya Maji et al. *YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss*. 2022. DOI: 10.48550/ARXIV.2204.06806.

[Ree+22] Reenul Reedha et al. "Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images." In: *Remote Sensing* 14.3 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14030592.

[Smi+19] Lyndon N. Smith et al. "Weed classification in grasslands using convolutional neural networks." In: *Applications of Machine Learning*. Edited by Michael E. Zelinski et al. Volume 11139. International Society for Optics and Photonics. SPIE, 2019, page 1113919. DOI: 10.1117/12.2530092.

[Sod+22] Sovi Guillaume Sodjinou et al. "A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images." In: *Information Processing in Agriculture* 9.3 (2022), pages 355–364. ISSN: 2214-3173. DOI: https://doi.org/10.1016/j.inpa.2021.08.003.

[Soz+22] Marco Sozzi et al. "Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms." In: *Agronomy* 12.2 (2022). ISSN: 2073-4395. DOI: 10.3390/agronomy12020319.

[Wan+21] J. Wang et al. "Deep High-Resolution Representation Learning for Visual Recognition." In: *IEEE Transactions on Pattern Analysis amp; Machine Intelligence* 43.10 (October 2021), pages 3349–3364. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2020.2983686.

[Wey+21] Jan Weyler et al. "Joint Plant Instance Detection and Leaf Count Estimation for In-Field Plant Phenotyping." In: *IEEE Robotics and Automation Letters* 6.2 (2021), pages 3599–3606. DOI: 10.1109/LRA.2021.3060712.

[WH21] Dandan Wang and Dongjian He. "Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning." In: *Biosystems Engineering* 210 (2021), pages 271–281. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2021.08.015.

[Wu+20] Xiaolong Wu et al. "Robotic weed control using automated weed and crop classification." In: *Journal of Field Robotics* 37.2 (2020), pages 322–340. DOI: https://doi.org/10.1002/rob.21938.

[Yan+23] Guoliang Yang et al. "A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention." In: *Agronomy* 13.7 (2023). ISSN: 2073-4395. DOI: 10.3390/agronomy13071824.

[Yu+19] Jialin Yu et al. "Deep learning for image-based weed detection in turfgrass." In: *European Journal of Agronomy* 104 (2019), pages 78–84. ISSN: 1161-0301. DOI: https://doi.org/10.1016/j.eja.2019.01.004.

[Zha+20] Xiaoguang Zhang et al. "A Unified Model for Real-Time Crop Recognition and Stem Localization Exploiting Cross-Task Feature Fusion." In: *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. 2020, pages 327–332. DOI: 10.1109/RCAR49640.2020.9303270.

[Zha+22] Jiawei Zhao et al. "Weed Detection in Potato Fields Based on Improved YOLOv4: Optimal Speed and Accuracy of Weed Detection in Potato Fields." In: *Electronics* 11.22 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11223709.

PAPER E

Zoom in on the plant: Fine-grained analysis of leaf, stem and vein instances

Ronja Güldenring[1,*], Rasmus Eckholdt Andersen[1] and Lazaros Nalpantidis[1]

[*]Corresponding author: ronjag@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

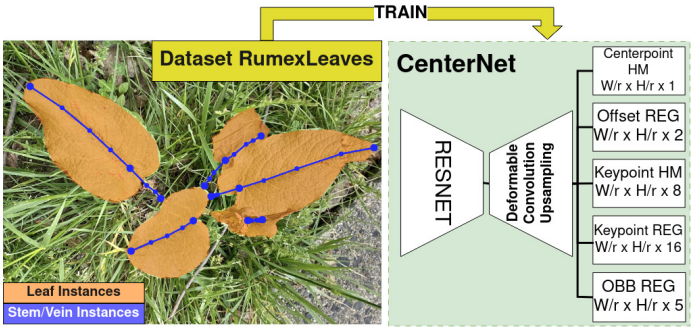
Abstract

Robot perception is far from what humans are capable of. Humans do not only have a complex semantic scene understanding but also extract fine-grained intra-object properties for the salient ones. When humans look at plants, they naturally perceive the plant architecture with its individual leaves and branching system. In this work, we want to advance the granularity in plant understanding for agricultural precision robots. We develop a model to extract fine-grained phenotypic information, such as leaf-, stem-, and vein instances. The underlying dataset *RumexLeaves* is made publicly available and is the first of its kind with keypoint-guided polyline annotations leading along the line from the lowest stem point along the leaf basal to the leaf apex. Furthermore, we introduce an adapted metric POKS complying with the concept of keypoint-guided polylines. In our experimental evaluation, we provide baseline results for our newly introduced dataset while showcasing the benefits of POKS over OKS.

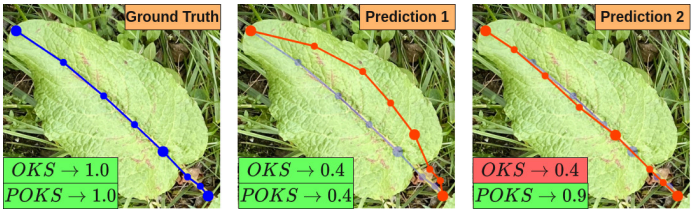
E.1 Introduction

In robotics, the ability to interact with the physical world is often limited by the granularity of the perception and understanding of the world. In precision agriculture, fine-grained visual analysis of individual plants can ultimately lead to the automation of labor- and physically demanding tasks, such as phenotyping or organic weed management. While the automation of phenotyping in outdoor agriculture has recently shown progress in the extraction of coarse characteristics, such as the leaf shape and size [Wey+22b; Wey+22a; Rog+23], finer detail analysis, such as detecting leaf veins, has not yet been addressed. Nevertheless, the length of the veins provides relevant insights on the plant growth state and health [Kon+21; Pan+22], whereas the course of the primary vein allows one to get a more detailed plant architecture with a leaf branching pattern. For weed management in crop fields, the common practice is to control weeds in early growth stages, when the joint stem is precisely identifiable. On the contrary, in grasslands, weeds only become visible in more advanced growing states when they stand out of their cluttered vegetation background. However, this comes at the cost of weeding complexity since joint stem positions are not clearly identifiable, but instead, multiple precise interventions at appropriately selected points on the weed are required.

In our work, we propose an approach that extracts simultaneously leaf instances and polylines for the stem and primary vein. The polyline is guided by keypoints, whose position is distinct, such as the lowest stem point, the leaf basal, and the leaf apex. In between those distinct keypoints, pseudo keypoints are spread evenly to indicate the course of the vein, however, their position is not distinct but may deviate along the line. In order to comply with the concept of *keypoint-guided polylines*, we propose an adapted metric *POKS*. As illustrated in Figure E.1b, *POKS* evaluates predictions that deviate from the line in accordance with the well-known *OKS* metric. On the contrary, when the keypoints deviate of same distance but along the line, *POKS* evaluates it with higher more reasonable scores.



(a) Our public dataset *RumexLeaves* includes leaf blade instance masks and keypoint-guided polylines for the stem and primary vein. We train a CenterNet model with custom heatmap (HM) and regression (REG) heads for different tasks.



(b) The new metric *POKS* evaluates keypoint predictions deviating from the line in accordance with the well-known OKS metric, as demonstrated in Prediction 1. However, when the keypoints deviate along the line as in Prediction 2 the metric based on *POKS* remains high.

Figure E.1: In this work we introduce (a) a new publicly available dataset *RumexLeaves* and (b) a new metric *POKS* which complies with keypoint-guided polylines.

We showcase our work in the domain of grassland, where we extract plant traits for a highly problematic weed *Rumex obtusifolius L.* [Gül+21; GEN23]. Compared to crop fields, grasslands constitute additional challenges with a more cluttered background, thin grass halms that can easily be confused with the stems as well as more occlusions from the background. Therefore, we claim that our work can be also considered for stem and vein extraction of crops of a similar nature, to further push forward the automation of phenotyping.

Our contributions of this work can be summarized as follows.

- We present an approach that allows for single-image fine-grained plant analysis—including stem and vein detection—in real-world settings, for the first time to the best of our knowledge. We make our code available at `https://github.com/DTU-PAS/RumexLeaves-CenterNet`.

- We introduce an adapted metric, the *POKS*, that complies with keypoint-guided polylines, consisting of distinct keypoints and pseudo keypoints, which may lie at any point on neighboring line segments.
- We make our fine-grained dataset *RumexLeaves* publicly available at `https://dtu-pas.github.io/RumexLeaves`, which includes 7747 manually labeled leaves with a pixel-wise annotation for the leaf blade and a polyline annotation, representing the corresponding stem and vein. To the best of our knowledge, it is the first real-world weed dataset containing stem and vein line annotations.

E.2 Related Work

E.2.1 Leaf vein extraction in laboratory settings

The extraction of leaf veins is an important property during plant phenotyping. The length and thickness of leaf veins give insights into the plant's growth state and health; the transportation capacity for water, nutrition, inorganic salts, and trace elements can be determined [Kon+21; Pan+22]. Since the manual measurement of leaf veins is a time-consuming and labor-intensive task, it has been of great interest to automate the process. In recent years, automatic leaf vein segmentation has been performed with classic computer vision approaches [Büh+15; ZW10; Li+11; Zhu+20], traditional machine-learning approaches with hand-crafted feature extraction [Sel+17; LH13; SCB21] as well as few deep learning approaches [Liu+22].

In all aforementioned works, experiments are performed on sharp high-resolution (or even microscope) images in laboratory settings with controlled lighting conditions.

E.2.2 Phenotyping in real-world settings

In real-world robotics settings, joint stem detection and leaf instance detection, i.e. leaf counting, are are primarily tackled for online plant phenotyping.

Leaf counting and instance segmentation in sugar beet crop fields have been extensively studied by Weyler et al. [Wey+21; Wey+22b; Wey+22a] and Roggiolani and Sodano et al. [Rog+23]. Recently, they made the corresponding dataset *PhenoBench* [Wey+23] publicly available, which includes plant and leaf instance annotations.

When plants have a fairly small size, joint stem detection enables automatic weeding, as for example in the work of Lottes et al. [Lot+18; Lot+20] and Weyler et al. [Wey+22b]. In our previous work [LGN23], we perform joint stem detection for *Rumex* plants. We found that even for the human annotator it was challenging to identify the precise joint stem position, especially in settings where plants have reached advanced growing states, their structure increases in complexity meaning the joint stems can not be clearly identified. Removing the weed gets more intractable due to the several keypoints required to fully remove them.

In our work, we aim to perform automated leaf vein extraction in real-world settings, which has not been explicitly addressed so far. While Marks et al. [MMS22] need to apply 3D reconstruction from multiple image views to extract advanced leaf traits, our approach only requires a single 2D image as input to solve the task of automatically extracting detailed plant traits. For *Rumex* weeds in real-world grassland fields, we prove that the identification of stems, primary veins, and corresponding individual leaves is feasible, advancing more fine-grained online phenotyping on robotics platforms.

E.3 Methods

E.3.1 CenterNet

The overall idea of CenterNet [ZWK19] is to identify object instances as keypoints. An encoder-decoder network structure is used to produce a heatmap, in which each peak in the heatmap represents a detected object. Additional properties are regressed wrt. to the center point position, such as e.g. the width and height of a corresponding 2D bounding box or keypoint positions relative to the center point. Hereby, image features at the center point position are considered for the regression. CenterNet has been proven to be a solid and robust object detection approach while being easily extendable to new custom regression tasks like e.g. instance segmentation [Wan+20] or multi-object tracking [Zha+21].

CenterNet performs regression on features at the spatial center point of objects. Thus, it can be infeasible to regress object properties that depend on features deviating spatially from the center point position. This especially applies to large and/or elongated objects. Deformable Convolutions (DC) [Zhu+18] increase the receptive field and allow the model to learn features spatially distant from the center point position. For DC, the applied image filter is not a fixed grid, but instead, the individual filter weights are applied with a learned offset. The offset depends on the input feature map and is therefore recalculated for each input tensor. The additional offset learning increases the number of model parameters and hence, the computational costs. However, common practice is to apply DC sparingly, leading to only a slight increase in model complexity.

More details about the various architectural settings will be elucidated in the analysis of section E.6.

E.3.2 Projected OKS

Object Keypoint Similarity (OKS) is a popular metric to evaluate the prediction of keypoints.

$$OKS = exp\left(-\frac{d_i^2}{2s^2\sigma_i^2}\right) \tag{E.1}$$

As shown in Equation E.1, for each keypoint, the Euclidean distance d_i is computed between prediction and target. d_i is normalized by the scale s which corresponds to the object size as well as σ, which represents the standard deviation between human annotations and true labels.

In this work, we aim to predict keypoint-guided polylines, which has *true keypoints* with distinct positions as well as *pseudo keypoints*, which are evenly spread between the *true keypoints* on a certain line, however, their position is not fully distinct and can vary along the line with one degree of freedom. In our evaluation metric, we want to comply with keypoint-guided polylines and allow the *pseudo keypoints* to deviate along the line to a certain extent. Therefore, we introduce a new metric, that we call *Projected OKS* (POKS).

While the *true keypoints* are still evaluated with the standard OKS, *pseudo keypoints* are projected to their two ground truth neighbor line segments. The projection serves as a temporary ground truth during the evaluation. As a concrete example, please refer to Figure E.2, where the stem/vein line of a leaf is described with three *true keypoints* $[k_{stem}, k_{basal}, k_{apex}]$ and five *pseudo keypoints* $[k_2, k_3, k_5, k_6, k_7]$. The red keypoints represent the predictions and are projected to the corresponding ground truth segments in blue; e.g. for the prediction of k_2 the closest projection is retrieved from the two line segments $[[k_{stem}, k_2], [k_2, k_3]]$. The projection is shown in yellow and is used as ground truth to compute the $mAP_{50:95,OKS}$. The projection is defined as follows

$$p(k_i, k_x) = k_i + t(k_i, k_x) \cdot (k_x - k_i) \tag{E.2}$$

$$t(k_i, k_x) = \text{clamp}\left(\frac{(\hat{y}_i - k_i) \cdot (k_x - k_i)}{(k_x - k_i) \cdot (k_x - k_i)}, 0, 1\right) \tag{E.3}$$

$$p_{min} = \min\left(||p(k_i, k_{i-1}) - \hat{y}_i||_2, p(k_i, k_{i+1}) - \hat{y}_i||_2\right) \tag{E.4}$$

$$POKS = exp\left(-\frac{p_{min}^2}{2s^2\sigma_i^2}\right) \tag{E.5}$$

where t is clamped to the range $[0, 1]$ to ensure the projected point falls within the line segment. Finally, the projected prediction p_{min} closest to the prediction $\hat{y}_i$ is used as ground truth to compute the OKS.

E.4 RumexLeaves Dataset

RumexLeaves contains in total 809 images with 7747 annotations of *Rumex obtusifolius* leaves. For each leaf, a pixel-wise leaf blade annotation as well as a polyline annotation of the stem — if visible — and vein were manually created. The stem is visible for $\approx 45\%$ of the leaves. We differentiate between two variants of datapoints:

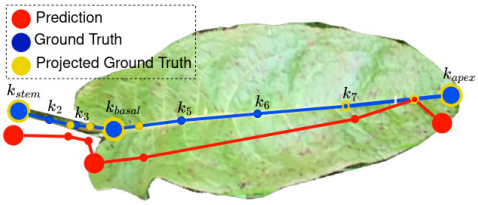


Figure E.2: The predicted keypoints (red) are projected to the corresponding ground truth neighbor line segments (blue). E.g. for the prediction of k_2 the closest projection of the two line segments $[[k_{stem}, k_2], [k_2, k_3]]$ is retrieved. Finally, the OKS is computed between the projected points (yellow) and the predictions (red). Please note, that the red predictions are not realistic, but instead, we came up with some extreme cases to demonstrate different projection cases.

(1) *iNaturalist* datapoints that have been downloaded from the plant publisher iNaturalist and (2) *RoboRumex* datapoints that have been collected with an agriculture robotics platform. Both variants originate from real-world settings and are specified further in the following subsections.

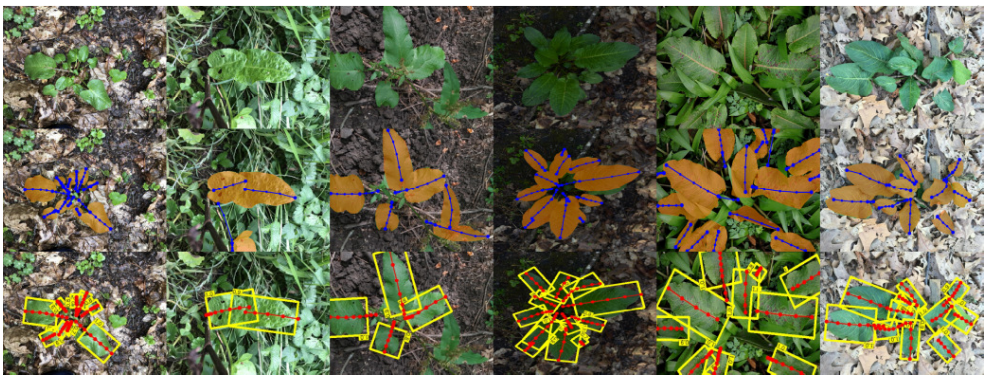
E.4.1 iNaturalist

iNatualist datapoints are 690 images that have been scraped from the plant image publisher iNaturalist [1]. We additionally provide a `references.txt`, that includes the expected attribution as requested by *CC-BY-4.0*. Since iNaturalist images are uploaded from people all over the world, the datapoints provide high variance in location, background and growing state. Figure E.3a provides examples for iNaturalist datapoints.

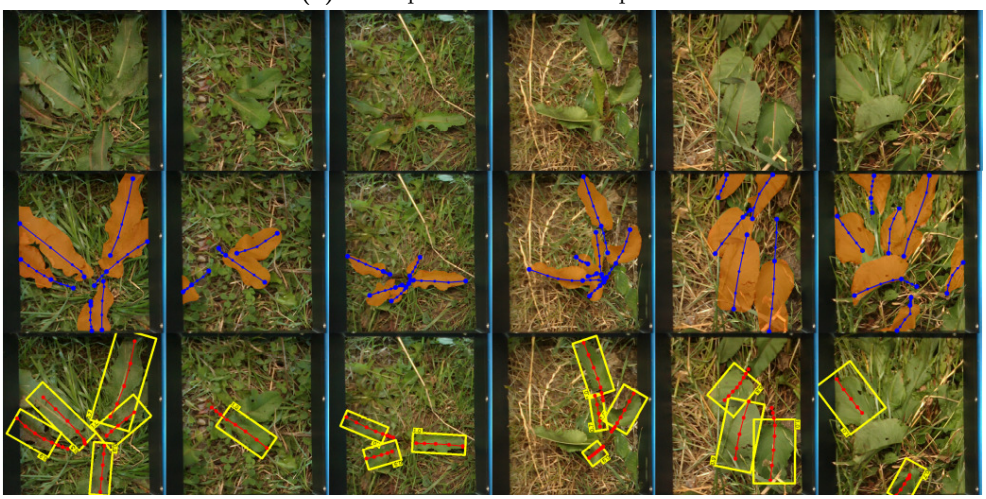
E.4.2 RoboRumex

119 RoboRumex datapoints have been collected with our robot platform at three different farms in the hinterland of Copenhagen in Denmark. The vision sensor *Intel-Realsense L515 Lidar* is used to collect the image data, providing RGB images as well as corresponding depth maps as shown in Figure E.3b. Moreover, each datapoint is accompanied by navigational information such as Odometry, GNSS, and IMU of the robot. For the data collection, we used a Clearpath Husky robot as shown in Figure E.4, which is specified in more detail in [GEN23]. The robot is equipped with two vision sensors: a forward-looking RGB camera in order to roughly detect the *Rumex* weeds [GEN23] and a close-up *IntelRealsense L515 LiDAR*. Once a weed plant is detected on the RGB camera imagery, the robot drives over the plant to perform a

[1] `https://www.inaturalist.org/`



(a) Example iNaturalist datapoints



(b) Example RoboRumex datapoints

Figure E.3: Example RGB images (top), overlayed annotations (middle), and final predictions (bottom). Ground truth annotations are shown in orange for the leaf instance masks and blue for the stem/vein polylines. Predictions are shown in yellow for leaf instance oriented bounding bocx and in red for the stem/vein polyline.

fine-grained analysis of the plant and treat it accordingly. The *L515* is mounted 0.5 m above ground pointing down through a hole in the robot chassis as shown in Figure E.4. Due to the *L515* sharing wavelengths with natural sunlight, it is mounted within a housing that provides controlled lighting conditions.

E.4.3 Image Annotations

In order to contain a high consistency in our *RumexLeaves* annotations, annotation rules have been defined beforehand.

In the following, we list the rules for the pixel-wise leaf blade annotation.
1. Generally, the polygon encloses the visible part of the *Rumex* leaf blade.
2. Exception to rule 1: When leaves are occluded in such a way, that the basal and apex of the leaf is visible, occlusions must be included in the annotation.
3. Minor occlusions, e.g. from grass and sticks, are ignored.
4. If the vein of the leaf is not visible because of unfavorable angles, the leaf is not annotated.

In the following, we list the rules for the stem and vein polyline annotation.
1. One polyline annotation represents the stem and the vein of one leaf with 8 keypoints: $[k_{stem}, k_2, k_3, k_{basal}, k_5, k_6, k_7, k_{apex}]$. Note, that it has 3 true keypoints $[k_{stem}, k_{basal}, k_{apex}]$ which have distinct positions. k_{stem} represents the lowest point of the stem, k_{basal} represents the lowest point of the vein at the leaf basal, and k_{apex} represents the highest point of the vein at the leaf apex. For the remaining pseudo keypoints, the hard requirement is to place them on the stem/vein line, while a soft requirement is to roughly spread them evenly between the true keypoints in order to serve as line guidance. The ambiguous definition of those pseudo keypoints introduces

Figure E.4: For data collection, the Clearpath Husky Robot has been used. It is equipped with navigational sensors and two vision cameras: an RGB camera to detect the weed and an IntelRealsense L515 Lidar to further analyze the weed in a close-up view. The L515 LiDAR is mounted 0.5 m above ground pointing downward through a hole in the robot. Furthermore, the black sensor housing provides controlled lighting conditions.

strong label noise and should be taken into account during evaluation.
2. If the stem is not visible, the polyline only consists of 5 keypoints: $[k_{basal}, k_5, k_6, k_7, k_{apex}]$.
3. There is no polyline annotation if the stem only is visible.
4. If the basal/apex of the leaf is occluded, the lowest/highest visible point of the vein is taken.

E.5 Experimental Setup

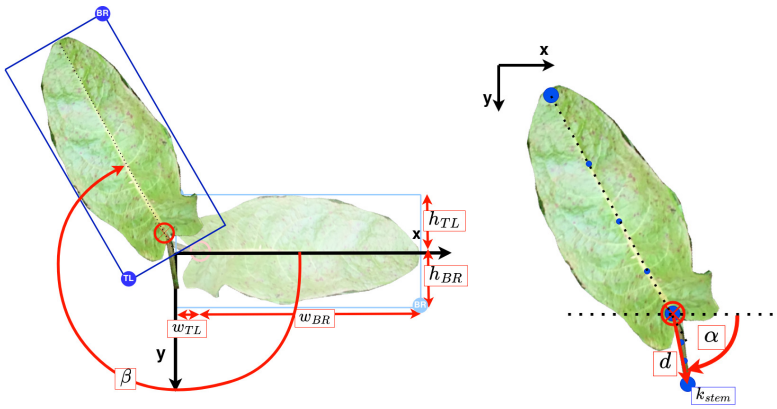
We develop a multi-task model, that predicts the stem/vein line as well as a rough estimate of the leaf size and orientation. However, during the evaluation, we set a stronger focus on the task of stem/vein line prediction, because it is less studied by the research community and our evaluation with POKS is one of our main contributions. Although the dataset provides segmentation masks for each leaf blade, we simplify the task and predict oriented bounding boxes (OBB), which can automatically be retrieved from the annotation: First, we use Principle Component Analysis on the vein keypoints to determine the orientation of the OBB, followed by the determination of box dimensions (width and height) on the zero oriented leaf segmentation.

E.5.1 Internal Leaf Coordinate System

During training, targets are provided in polar coordinates, because it aligns well with the regression of OBBs. For each keypoint, the distance d and angle α to the center point are given as demonstrated in Figure E.5b for k_{stem}. For the OBB prediction, a straightforward approach would be to predict the width and height as well as the smallest angle $\beta < 90$ř, which points either toward the leaf's axis or perpendicular to its axis. However, we want to force the model to learn leaf hierarchies, i.e. the location of basal and apex. As shown in Figure E.5a, we start from the zero-oriented position where the apex of the leaf is pointing in the direction of the x-axis of the image plane coordinate system. β is the angle between the leaf in the default position and the leaf in the true position. The width of the bounding box corresponds to the length of the leaf, while the height of the bounding box corresponds to the width of the leaf. Since the center point is not necessarily representing the center of the bounding box, but rather the object, width and height to the top-left and bottom-right corner are provided: $w = w_{TL} + w_{BR}$, $h = h_{TL} + h_{BR}$.

E.5.2 Metrics

We evaluate the keypoint-guided polyline with both the original OKS and the POKS introduced to show the difference. In order to get an in-depth analysis of the model performance, we report the *POKS* for different parts of the polyline: **All** $[k_{stem}, k_2, k_3, k_{basal}, k_5, k_6, k_7, k_{apex}]$, **Stem** $[k_{stem}, k_2, k_3, k_{basal}]$,



(a) OBB target **(b)** Keypoint target

Figure E.5: Keypoint and OBB targets are defined wrt. to the center point which is marked with the red circle at the leaf's basal. Left: The OBB dimensions are taken from the leaf's default position where the leaf tip is pointing along the x-axis of the image coordinate system. The angle β between the leaf in the default position and in the true position is the orientation of the OBB. Right: Keypoint targets are represented with the distance d and angle α wrt. the center point.

Vein $[k_{basal}, k_5, k_6, k_7, k_{apex}]$, **True** $[k_{stem}, k_{basal}, k_{apex}]$, **Pseudo** $[k_2, k_3, k_5, k_6, k_7]$.

For the evaluation of the oriented Bounding Box (OBB) prediction, we consider the official implementation[2] of the DOTA dataset [Xia+17], which is a large-scale benchmark dataset for oriented object detection in aerial images. The implementation is based on the original mAP_{50} VOC implementation for straight bounding boxes.

E.5.3 Model Design

In Figure E.1a, we give an overview of the model architecture with all potential heatmap (HM) and regression (REG) heads. The architecture has a shared encoder-decoder block, consisting of the ResNet architecture as backbone, followed by three upsampling blocks with 256, 128, and 64 output channels respectively. On top of that, multiple heads are considered during our experimental evaluation as described in the following.

- Centerpoint heatmap $[W/r \times H/r \times 1]$: Each peak in the heatmap represents a detected leaf. The output is downscaled by ratio r.

[2] https://github.com/CAPTAIN-WHU/DOTA_devkit/blob/master/dota-v1.5_evaluation_task1.py

- Offset regression $[W/r \times H/r \times 2]$: The identified center point position of each leaf is corrected by the offset in order to comply to the down-scaled output map.
- Keypoint regression $[W/r \times H/r \times 16]$: 8 keypoints (d, α) are regressed wrt. the corresponding centerpoint.
- Keypoint heatmap $[W/r \times H/r \times 7/8]$: The keypoint prediction is refined by assigning each regressed keypoint location to the closest keypoint in the heatmap [ZWK19]. We consider two different types of heatmaps: P-heatmap and S-heatmap. P-heatmap has 8 channels and is a Gaussian-rendered heatmap based on the keypoint position. S-heatmap has 7 channels is a Gaussian rendered heatmap based on the polyline segments, i.e. $[k_{stem}, k_2], [k_2, k_3], ..., [k_7, k_{apex}]$.
- OBB regression $[W/r \times H/r \times 3/5]$: Either 3 (w, h, β) or 5 $(w_{TL}, w_{BR}, h_{TL}, h_{BR}, \beta)$ are regressed wrt. the corresponding centerpoint.

The final loss consists of a weighted sum of L1 loss for regression heads and focal loss for dense object detection [Lin+17] for heatmap heads as follows

$$\mathcal{L}_{total} = w_{cp}\mathcal{L}_f + w_{off}\mathcal{L}_1 + w_{kp}\mathcal{L}_1 + w_{kphm}\mathcal{L}_f + w_{obb}\mathcal{L}_1$$

, where $w_{cp} = 1.0$, $w_{off} = 1.0$, $w_{kp} = 20.0$, $w_{kphm} = 1.0$, $w_{obb} = 20.0$.

Note, that we do not apply any post-processing of predictions, such as restricting the predicted keypoints to the boundaries of the predicted OBB.

E.5.4 Fixed Hyperparameters

For all experiments, we fix a number of training settings. During our ablation studies, we only consider the iNaturalist images from *RumexLeaves*, which are split randomly into train, val, and test by 0.7/0.15/0.15. The input size is fixed to $(512, 512)$ to ensure that the stems of the leaves are represented with a large enough amount of pixels to be distinguishable and identifiable. All models are trained with a batch size of 32 for 96000 iterations, which corresponds to 6001 epochs for the iNaturalist train split. The network is updated with the Adam optimizer, with multistep learning scheduling applied. The initial learning rate is at 0.0005 and is updated at epoch $[2500, 4000, 5000]$ with a constant $\gamma = 0.5$. During training, standard augmentation is applied to the image data: random flip, zoom out, color jitter, rotate by 90°, random brightness, random contrast. Furthermore, the images are normalized. The model backbone is initialized with pre-trained ImageNet weights.

E.6 Experimental Results

In section E.6.1, we perform ablation studies for each task – keypoint and OBB prediction – individually. In section E.6.2, we fix the settings to the best results of the

ablation studies and train a model in a multi-task setting. The best-performing model is applied to the test data split in section E.6.3, accompanied by some qualitative prediction examples.

E.6.1 Ablation Studies

E.6.1.1 Target Design

In this ablation study, the target design for the keypoint prediction only is examined in Table E.1.

In the first two experiments, we elaborate on the angle representation of the keypoint prediction. In experiment 0, the angle is predicted as normalized angle α in the range of $[0, 1]$. In experiment 1, we use the representation $cos(\alpha), sin(\alpha)$ which ensures continuity throughout the whole angle range. The value pair can be decoded back to $\alpha = arctan(sin(\alpha), cos(\alpha))$. The results in Table E.1 show, that experiment 1 with continuous angle representation improves over experiment 0. Therefore, we fix this representation for the following experiments.

In the original CenterNet [ZWK19] implementation, the center point is represented by the center of the object's bounding box. In order to increase the focus on the stem and vein line, the true keypoints on the polyline are considered as center points to represent a leaf. Experiments 2 – 5 in Table E.1 show the results for the different center points and using k_{basal} as center point clearly outperforms its alternatives. Therefore, we use k_{basal} as center point for all following experiments.

Experiments 6 – 7 in Table E.1 show that introducing heatmap refinement – via P-heatmap or S-heatmap – leads to some improvements. However, the difference between P-heatmap and S-heatmap is not crucial.

As expected, the $mAP_{50:95,POKS}$ performs better than the original $mAP_{50:95,OKS}$ due to the looser projection criterion. The performance of the stem prediction is clearly lower than the one of the veins. Stems are often camouflaged in the background, which has similar characteristics such as grass stalks or small sticks. Moreover, stems contain only a small amount of pixels, which decreases the number of

Table E.1: Ablation study for different target designs.

	Target		$mAP_{50:95,POKS}$					$mAP_{50:95,OKS}$
ID	Which target?	Specification	All	Stem	Vein	True	Pseudo	
0	angle	angle/360	0.354	0.280	0.448	0.167	0.481	0.186
1		cos(angle), sin(angle)	0.378	0.275	0.491	0.175	0.514	0.196
2	center point	center of OBB	0.313	0.165	0.474	0.132	0.442	0.146
3		kp_{stem}	0.287	0.279	0.299	0.133	0.405	0.150
4		kp_{basal}	0.378	0.275	0.491	0.175	0.514	0.196
5		kp_{apex}	0.320	0.130	0.505	0.145	0.436	0.138
6	keypoints	regressed only	0.378	0.275	0.491	0.175	0.514	0.196
7		regressed + P-heatmap	0.382	0.291	**0.493**	0.182	**0.521**	0.203
8		regressed + S-heatmap	**0.385**	**0.302**	0.483	**0.185**	0.519	**0.208**

available features. One could argue, that veins are as thin as stems and are therefore represented by the same number of features. However, veins are embedded in the leaf blade which indirectly inflates the number of features, leading to a better prediction performance.

E.6.1.2 Model Complexity

We take the target setup from experiment 4 in Table E.1 and vary the model complexity in Table E.2. In experiment 0, a model with a ResNet18 backbone and two Convolutional Layers for the keypoint head is used. When looking at the model predictions qualitatively, it is striking that keypoint predictions close to the center point are more accurate than those that deviate from the center point. The reason for a decreasing performance with increasing distance from the center point is a potentially too small receptive field that is not capable of taking into account features further away from the center point. In experiment 1, we increase the receptive field by increasing the number of layers. This action has minimal impact on the performance of the network. In experiment 2, the Convolutional Layers in the up-sampling blocks of the ResNet are replaced with Deformable Convolutions, leading to a relevant performance boost with 2.8% improvement for the $mAP_{50:95,POKS}$ and 6% improvement for the original $mAP_{50:95,OKS}$. Finally, we replace the Convolutional Layers in the keypoint head with Deformable Convolution, leading to further improvement. The best model is a Resnet backbone with Deformable Convolutions in the upsampling blocks as well as two Deformable Convolutions in the keypoint head as listed in experiment 3. In experiments 5 and 6, the backbone complexity is further increased by applying deeper models, such as ResNet34 and ResNet50, resulting in further improvement. The best performing model is the ResNet50 variant with a $mAP_{50:95,POKS}$ of 0.460 and a $mAP_{50:95,OKS}$ of 0.267. We apply the same ablation study of model complexities for OBB detection only by introducing Deformable Convolutions in the upsampling blocks of the ResNet as well as in the OBB head. The results are shown in Table E.3 and the effect of Deformable Convolutions is not as strong, yet visible. The best-performing variant – ResNet18 + DC and 3 Deformable Convolutions in the OBB head – is further used with deeper ResNet variants.

Table E.2: Ablation study of model complexities for keypoint detection.

ID	Model Complexity		$mAP_{50:95,POKS}$					$mAP_{50:95,OKS}$
	Backbone	KP Head	All	Stem	Vein	True	Pseudo	
0	ResNet18	2 Conv	0.350	0.282	0.432	0.136	0.500	0.170
1	ResNet18	3 Conv	0.352	0.292	0.429	0.138	0.501	0.179
2	ResNet18 + DC	2 Conv	0.378	0.275	0.491	0.175	0.514	0.196
3	ResNet18 + DC	2 DC	0.405	0.301	0.522	0.219	0.528	0.230
4	ResNet18 + DC	3 DC	0.397	0.295	0.514	0.217	0.517	0.224
5	ResNet34 + DC	2 DC	0.433	0.325	0.549	0.237	0.556	0.242
6	ResNet50 + DC	2 DC	**0.460**	**0.366**	**0.563**	**0.259**	**0.586**	**0.267**

Table E.3: Ablation study of model complexities for OBB detection.

ID	Backbone	OBB head	$mAP_{50,OBB}$
0	ResNet18	2 Conv	0.705
1	ResNet18	3 Conv	0.706
2	ResNet18 + DC	2 Conv	0.709
3	ResNet18 + DC	2 DC	0.710
4	ResNet18 + DC	3 DC	<u>0.712</u>
5	ResNet34 + DC	3 DC	0.714
6	ResNet50 + DC	3 DC	**0.776**

E.6.2 Multi-Task Performance

We apply the best performing model 6 from Table E.2 to a multi-task setting where both tasks - keypoint and OBB detection – are solved simultaneously. Results are listed in Table E.4. We used the loss weights $w_{obb} = 10.0$ and $w_{kp} = 10.0$. Though the individual models marginally outperform the multi-task model, the performance drop is not large enough to justify the increased computational complexity.

E.6.3 Final Results

The multi-task model from the previous section serves as our final model. We apply the final model to the iNaturalist as well as the RoboRumex datapoints and list the results in Table E.5. Note, that the model is trained on 70% of the iNaturalist datapoints only. The model performs on par on the iNaturalist test split, but much worse on the RoboRumex test split. We evaluate on the RoboRumex test split only to allow comparisons with future developments, that potentially include RoboRumex data during training. Although the Rumex leaves in the iNaturalist datapoints are diverse with a range of different backgrounds and different leaf shapes as well as appearances, the model is not performing sufficiently on a new data distribution such as the one collected with the robot. Reasons for the lower performance can be the image quality/sharpness as well as the predominantly occurrence of small leaves. In Figure E.3a and E.3b, we show the prediction vs. ground truth for sample images for the iNaturalist and RoboRumex test split datapoints respectively.

Table E.4: Multi-task performance wrt. to their single task performance (↓).

$mAP_{50:95,POKS}$					$mAP_{50:95,OKS}$	$mAP_{50,OBB}$
All	Stem	Vein	True	Pseudo		
0.446 (↓ 0.014)	0.347 (↓ 0.019)	0.559 (↓ 0.004)	0.246 (↓ 0.013)	0.574 (↓ 0.012)	0.256 (↓ 0.011)	0.709 (↓ 0.067)

Table E.5: Final performance on test split of iNaturalist and RoboRumex datapoints.

ID	Test Split	$mAP_{50:95,POKS}$					$mAP_{50:95,OKS}$	$mAP_{50,OBB}$
		All	Stem	Vein	True	Pseudo		
0	iNaturalist	0.450	0.313	0.600	0.246	0.579	0.265	0.716
1	RoboRumex	0.244	0.171	0.390	0.080	0.406	0.109	0.669

E.7 Conclusion

We present a fine-grained plant analysis in a real-world grassland setting. We train a custom CenterNet model to extract each leaf instance with an oriented bounding box and the corresponding stem and vein polyline. The performance is evaluated with standard metrics such as OKS and mAP as well as an adapted keypoint evaluation *POKS*, which complies with keypoint-guided polylines, consisting of distinct as well as pseudo keypoints. Additionally, we publish the corresponding dataset *RumexLeaves* with 7747 manually annotated leaves at https://dtu-pas.github.io/RumexLeaves to encourage other researchers to improve on our results.

E.8 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258. The authors would like to thank the organic farmers Erik Hansen (Lundholm), Allan Clausen (Hegnstrup), and Otto Stengaard (Stengaard) for offering their fields for data collection and insightful discussions regarding Rumex treatment.

References

[Büh+15] Jonas Bühler et al. "phenoVein—A Tool for Leaf Vein Segmentation and Analysis ." In: *Plant Physiology* 169.4 (October 2015), pages 2359–2370. ISSN: 0032-0889. DOI: 10.1104/pp.15.00974.

[GEN23] Ronja Güldenring, Frits K. van Evert, and Lazaros Nalpantidis. "RumexWeeds: A grassland dataset for agricultural robotics." In: *Journal of Field Robotics* (2023). DOI: https://doi.org/10.1002/rob.22196.

[Gül+21] Ronja Güldenring et al. "Few-leaf learning: Weed segmentation in grasslands." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pages 3248–3254.

[Kon+21] Tukhar Jyoti Konch et al. "Remarkable Rate of Water Evaporation through Naked Veins of Natural Tree Leaves." In: *ACS Omega* 6.31 (2021), pages 20379–20387. DOI: 10.1021/acsomega.1c02398.

[LGN23] Jiahao Li, Ronja Güldenring, and Lazaros Nalpantidis. "Real-Time Joint-Stem Prediction for Agricultural Robots in Grasslands Using Multi-Task Learning." In: *Agronomy* 13.9 (2023). ISSN: 2073-4395. DOI: 10.3390/agronomy13092365.

[LH13] Kuebum Lee and K.-S Hong. "An implementation of leaf recognition system using leaf vein and shape." In: *International Journal of Bio-Science and Bio-Technology* 5 (January 2013), pages 57–65.

[Li+11] C. Li et al. "Extraction of leaf vein based on improved Sobel algorithm and hue information." In: *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering* 27 (July 2011), pages 196–199. DOI: 10.3969/j.issn.1002-6819.2011.07.034.

[Lin+17] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection." In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pages 2999–3007. DOI: 10.1109/ICCV.2017.324.

[Liu+22] Xiaobao Liu et al. "Plant leaf veins coupling feature representation and measurement method based on DeepLabV3+." In: *Frontiers in Plant Science* 13 (2022). ISSN: 1664-462X. DOI: 10.3389/fpls.2022.1043884.

[Lot+18] Philipp Lottes et al. "Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming." In: June 2018. DOI: 10.1109/IROS.2018.8593678.

[Lot+20] Philipp Lottes et al. "Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming." In: *Journal of Field Robotics* 37.1 (January 2020), pages 20–34. ISSN: 1556-4959. DOI: 10.1002/rob.21901.

[MMS22] Elias Marks, Federico Magistri, and Cyrill Stachniss. "Precise 3D Reconstruction of Plants from UAV Imagery Combining Bundle Adjustment and Template Matching." In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pages 2259–2265. DOI: 10.1109/ICRA46639.2022.9811358.

[Pan+22] Ling Pan et al. "Coordination of stomata and vein patterns with leaf width underpins water-use efficiency in a C4 crop." In: *Plant, Cell & Environment* 45.6 (2022), pages 1612–1630. DOI: https://doi.org/10.1111/pce.14225.

[Rog+23] Gianmarco Roggiolani et al. "Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain." In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. 2023.

[SCB21] Guruprasad Samanta, Amlan Chakrabarti, and Bhargab Bhattacharya. "Extraction of Leaf-Vein Parameters and Classification of Plants Using Machine Learning." In: January 2021, pages 579–586. ISBN: 978-981-15-7833-5. DOI: 10.1007/978-981-15-7834-2_54.

[Sel+17] Jesse Dave S. Selda et al. "Plant Identification by Image Processing of Leaf Veins." In: *Proceedings of the International Conference on Imaging, Signal Processing and Communication*. ICISPC 2017. Penang, Malaysia: Association for Computing Machinery, 2017, pages 40–44. ISBN: 9781450352895. DOI: 10.1145/3132300.3132315.

[Wan+20] Yuqing Wang et al. "CenterMask: Single Shot Instance Segmentation With Point Representation." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pages 9310–9318. DOI: 10.1109/CVPR42600.2020.00933.

[Wey+21] J. Weyler et al. "Joint Plant Instance Detection and Leaf Count Estimation for In-Field Plant Phenotyping." In: 6 (2 2021), pages 3599–3606. DOI: 10.1109/LRA.2021.3060712.

[Wey+22a] J. Weyler et al. "In-Field Phenotyping Based on Crop Leaf and Plant Instance Segmentation." In: 2022.

[Wey+22b] J. Weyler et al. "Joint Plant and Leaf Instance Segmentation on Field-Scale UAV Imagery." In: 7.2 (2022), pages 3787–3794. ISSN: 377-3766. DOI: 10.1109/LRA.2022.3147462.

[Wey+23] Jan Weyler et al. "PhenoBench — A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain." In: *arXiv preprint* (2023).

[Xia+17] Gui-Song Xia et al. "DOTA: A Large-scale Dataset for Object Detection in Aerial Images." In: *CoRR* abs/1711.10398 (2017).

[Zha+21] Yifu Zhang et al. "Fairmot: On the fairness of detection and re-identification in multiple object tracking." In: *International Journal of Computer Vision* 129 (2021), pages 3069–3087.

[Zhu+18] Xizhou Zhu et al. "Deformable ConvNets v2: More Deformable, Better Results." In: *CoRR* abs/1811.11168 (2018).

[Zhu+20] Jiyou Zhu et al. "A Fast and Automatic Method for Leaf Vein Network Extraction and Vein Density Measurement Based on Object-Oriented Classification." In: *Frontiers in Plant Science* 11 (2020). ISSN: 1664-462X. DOI: 10.3389/fpls.2020.00499.

[ZW10] Xiaodong Zheng and Xiaojie Wang. "Leaf Vein Extraction Based on Gray-scale Morphology." In: *International Journal of Image, Graphics and Signal Processing* 2 (December 2010). DOI: 10.5815/ijigsp.2010.02.04.

[ZWK19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as Points." In: *CoRR* abs/1904.07850 (2019).

PAPER F

Smart implements for robotic precision agriculture: A case study on laser and electric weeding in grasslands

Ronja Güldenring[1,*], Jacobo Domínguez[2], Ayodeji O. Deolu-Ajayi[3], Menno Sytsma[4], Esther López[2], Julien Fradin[5], Hans Pekkeriet[6], Harry Koonstra[7], Annelies Beniers[3], Anouk Leunissen[4], Ard Nieuwenhuizen[3], Lazaros Nalpantidis[1], Frits K. van Evert [3]

[*]Corresponding author: ronjag@dtu.dk
[1]Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
[2]Acorde Technologies, 39011 Santander, Spain
[3]Agrosystems Research, Wageningen University & Research, PO Box 16, 6700 AA Wageningen, The Netherlands
[4]Greenhouse Horticulture & Flower Bulbs, Wageningen University & Research, PO Box 644, 6700 AA Wageningen, The Netherlands
[5]IDELE – Institut de l'Elevage, 63170 Aubière, France

[6]Pekkeriet Dalfsen B.V., 7722 KK Dalfsen, The Netherlands
[7]Loonbedrijf Koonstra, 7739 PA Vinkenbuurt, The Netherlands

Abstract

Field operations in arable and grassland farming are typically done with farming implements (sprayers, weeders, plows, harrows, etc.) that are towed or carried by a tractor. A relatively new development is that a "smart" implement assumes some control over the tractor (such as speed). Agricultural robots, however, are typically still single-purpose machines and if they use implements, do not allow them to take control. We applied the concept of a vehicle and a smart implement to robotic weeding of broad-leaved dock in organic dairy farming. We present two smart implements that can be mounted to the same robotic vehicle, one for laser weeding and one for electric weeding. The implements contain sensors and software to detect Rumex weeds, as well as software to control all aspects of the movement of the robotic vehicle. The laser weeding robot has been evaluated in practice and functions well. The electric weeding implement needs further work before it can be evaluated in practice. Our results indicate that the concept of a smart implement that controls a robotic vehicle is viable and holds promise for the future of agricultural robotics.

F.1 Introduction

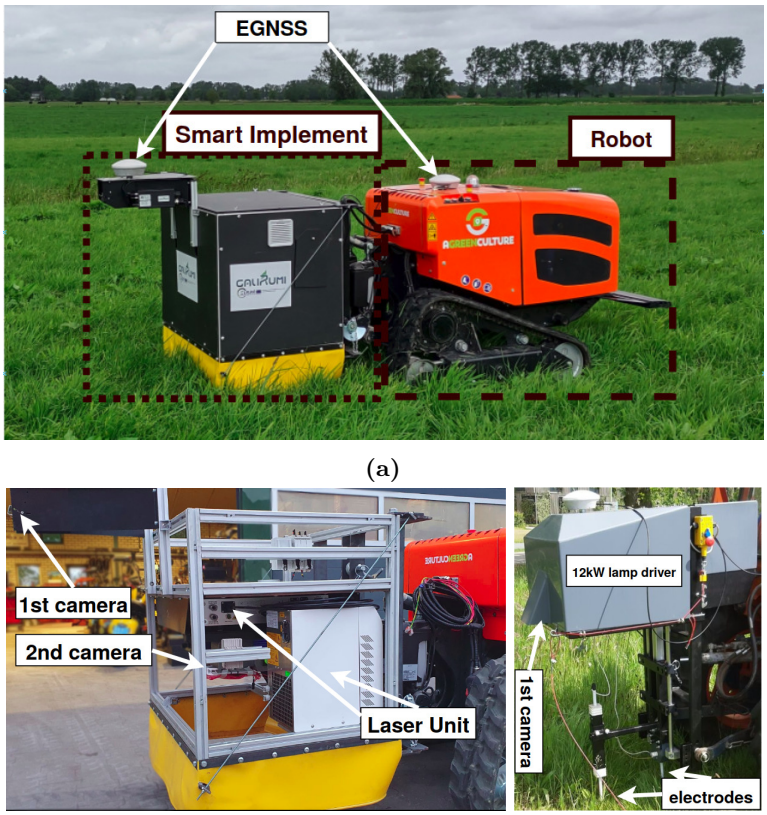
Agriculture provides humans with necessary food and raw materials, but its environmental impact is unacceptably high. Much effort is underway to make agriculture more sustainable, for example through precision agriculture or organic agriculture. The development of mobile autonomous systems for precision agriculture has advanced greatly in the last few decades, including applications such as weed management [Wey+22; GEN23; Ols+19] to harvesting [Par+; Bir+20], seeding/planting [Ble+16], and monitoring [Cub+20]. The research projects have in many cases developed customized robot vehicles specifically for their individual tasks. Those robots serve as prototypes to prove the concept of proposed state-of-the-art approaches. However, from a commercial point of view, it is infeasible to purchase multiple robots, each for a different task. Field operations in arable and grassland farming are typically done with farming implements (sprayers, weeders, plows, harrows, etc.) that are towed or carried by a tractor. The tractor in turn is steered by the driver, who navigates the tractor through the field, switches the implements on and off, and monitors the quality of the work. It is attractive to use this concept also in robotic farming. In robotic farming, there is no driver to perform these tasks. In principle a task such as navigation can be performed by the robotic vehicle, however, it may be that the implement imposes specific requirements on navigation, and a high degree of interaction will have to take place between the implement and the vehicle. Here we propose to realize the intelligence fully to the implement. This

concept is in the process of being introduced to conventional tractors and implements (Tractor Implement Management (TIM), https://www.aef-online.org/about-us/activities/tractor-implement-management-tim.html) but has so far not been described for agricultural robots. We envision a modular agricultural robot setup: The base platform is a commercial and matured robot, that can be hooked up with different smart implements to perform various tasks on a farm. The base robot is of reasonable size and price, thus, will be attractive to dairy farmers who do not have access to a farm contractor, have smaller farms, or wish to own and operate their own robot. The farmer owns a small robot that he/she can deploy in combination with the appropriate smart implement on a pasture whenever it fits into the work schedule.

We selected weeding as the use case for our work. In all of agriculture, controlling weeds is a major issue. In dairy farming, broad-leaved dock (*Rumex obtusifolius L.*) is a problematic weed. If left uncontrolled, it will expand, cover large parts of the farm, and significantly reduce grass yield. In conventional dairy farming, the weed can be kept in check with repeated application of herbicides. In organic dairy farming, a suitable weed control method is not available. Conventional dairy farmers are reluctant to switch to organic farming because of broad-leaved dock. Organic dairy farming will only increase substantially when an efficient weed control method is available. Attention has been given to the detection of broad-leaved docks [BF17; HIR+12; KTN18; ŠKA10; Eve+09; Gül+21; GEN23; LGN23] as well as to robotic weeding of broad-leaved docks [Lat+07; Eve+11; Lat+17; LS14; Pal23].

Many weed control methods have been tested for control of broad-leaved dock, including microwaves [Lat+07], chopping [Eve+11], treatment with a hot steel rod [RK07], and the application of hot water to the roots [Lat+17; LS14]. These methods all suffer from limitations, including expense, heavy equipment, frequent breakdown, and large energy use. We will use two techniques – laser weeding and electric weeding – that have so far not been used to control broad-leaved dock, and which can be achieved with a reasonable robot size. Laser weeding has been applied primarily to seedlings [Mar+12; Hei+01; Col+21]. When applied to Rumex it works by removing the leaves of the weed several times throughout the growing season. When done often enough, this will eventually exhaust the reserves in the tap root [Eve+20]. Electric weeding works by guiding an electric current through the weed, which kills or seriously damages the entire plant [DIP+85; Col+19; VB01].

In this work, we present two smart implements that can be mounted to the same robot base platform, one for laser weeding and one for electric weeding. We introduce and evaluate the individual components that go into the two implements, which constitute the vision system, the robot EGNSS localization, the robot navigation as well as the laser weeding unit, or the electric weeding unit. Finally, we present initial results of the modular robotics system by integrating the laser implement into the robot base platform and applying it to real-world *Rumex* fields in three different locations in Europe. Due to time constraints, the electric implement was not tested on the robot, instead, it was mounted to a tractor (using the same three-point linkage that is on the robot) to also show its performance on a real-world *Rumex* field. Positive



(b) Laser implement **(c)** Electric implement

Figure F.1: (a) The base platform is the commercial CEOL robot from AGreenCulture. Two smart attachable implements have been developed: (b) a laser implement and (c) an electric implement. Both implements use the same vision and localization system, but differ in the applied weeding technique.

findings from both weeding implement prototypes point towards a promising future, with subsequent actions already in progress. One might consider our demonstration video for a visual summary of our work https://youtu.be/3LOMdMfIjOs.

F.2 System requirement

To identify system requirements, meetings and conversations were held with farmers and farm contractors. We present identified system requirements for the base robot platform (subsection F.2.1) and the implement implementation (subsection F.2.2) in general and finally, system requirements for the use case of herbicide-free *Rumex* weeding in Grasslands in subsection F.2.3.

F.2.1 General Robot Requirements

The robot must be able to work in dry weather conditions during a regular growth season. It would be an advantage if it could also work in the rain. The robot must be of a size that it is transportable from field to field within a normal van or trailer behind the car. Therefore, it is expected that the robot can drive up ramps, preferably autonomously, else remotely controlled. Electrical operation is preferred over fuel due to lower cost and environmental-friendly operation, which is an important factor for organic farms.

F.2.2 General Smart Implement Requirements

The smart implement is expected to be easily integrable. Mounting points must be intuitive and feasible to mount with standard tools. A setup should not take longer than 30 min. This assures flexible implement integration and reduces the obstacle of exchanging between different implements. The implement is expected to not exceed a weight of 100kg and must be of a size so that it is transportable within the same van as the robot. In accordance with the robot requirements, the implement must be able to work in all weather conditions during a regular growth season.

F.2.3 *Rumex* Weeding Implement

The smart weeding implement is supposed to identify *Rumex* weeds while driving. Once a weed is detected, the implement controls the robot to stop for herbicide-free weeding. The implement must be able to work in normal daylight conditions. It would be an advantage if it could also work in darkness/night conditions. It is preferred that the weed is destroyed within a single treatment. However, if that is not feasible, it is expected that the *Rumex* plant locations are saved within a map. In all following weeding iterations of the year, the robot is expected to only target registered plants and not scout the whole field again. The field with weed locations and treatments

must be provided as a map to the robot manager. The implement-robot integration is expected to work at least one day, i.e. 8 hours without break, with a minimum working rate of 1 ha per day. Recharging or filling up the fuel tank is expected to happen at night. The implement is expected to identify static and dynamic obstacles, e.g. fence posts, cows, and other robots. The implement is expected to control the robot to avoid collisions with obstacles. The robot is expected to notify the robot manager on unexpected occasions such as when the robot is stuck in the field, the battery is low, or the emergency button is pressed. It is also expected to give a notification once it has finished weeding the field.

F.3 System architecture

F.3.1 Hardware components

F.3.1.1 Robotic vehicle

We selected the robot from AgreenCulture [1]. This robot is a tracked vehicle that weighs 800 kg, is powered by a 12.7 kW diesel engine, and can operate continuously for 10 hours on its 30 L fuel tank. The robot is driven by 48 V DC electric motors. A single electronic unit, the AGCBox, contains an RTK-GNSS receiver as well as the guiding software. The robot is equipped with bumpers for obstacle detection. A three-point linkage [2] allows using standard agricultural implements.

F.3.1.2 Vision system

For the vision system, two commercially available cameras are considered: (1) a first-stage camera to be used when the robot is searching for weeds, and (2) a second-stage camera to be used when the robot has come to a stop close to the weed and a close-up analysis of the detected weed must be made. Images from both cameras are processed by a Jetson Xavier NX, which has a 6-core NVIDIA Carmel Arm processor, a 384-core NVIDIA Volta GPU, and 8 GB of RAM.

First-stage camera The first-stage camera is a Basler ace 2 and has a global shutter with an image resolution of 1920 px $\times$ 1200 px and pixel size of 3.45 $\mu m \times$ 3.45 μm, while it is connected via USB3.0. The mounting position of the first-stage camera is a trade-off between field coverage area and weed detection performance. The higher the camera is mounted, the higher the field coverage, but the smaller objects get in the imagery. Small object detection is represented with a small number of pixels, which do not contain enough features to be identified robustly. We mount the Basler ace 2 camera at a height of 1m, tilted 75° toward the ground. The setup

[1] https://www.agreenculture.net/en/ceol
[2] https://www.iso.org/standard/41233.html

results in a field coverage of minimum 1.5 m width and small leaves (3cm × 3cm) are represented with minimum ≈ 838px.

Second-stage camera The second-stage camera is an Intel RealSense Lidar Camera L515. It provides RGB as well as depth images of the weed, that has been initially detected with the first-stage camera. The sensor is mounted 0.5 m above ground pointing straight down to the ground. The sensor is mounted in controlled lighting conditions, i.e. under a shield with LED lightning, because the wavelength of natural light interferes with the wavelength of the L515 sensor and reduces the accuracy of the depth map. The L515 imagery is used to perform a fine-grained analysis of the weed plant.

F.3.1.3 Laser Weeding

Rumex can be controlled (if perhaps not killed) by frequent defoliation. Frequent defoliation exhausts the weed and reduces the impact on grass yield below the damage threshold, it limits the proliferation of the weed, and it may reduce prevalence [Eve+20]. Using a laser to defoliate Rumex has the advantage that there is no physical contact with either plant or soil; this eliminates the wear and breakdowns that would result from physical contact.

The first laser was built in 1960 and now a broad range of laser modules is available. Lasers can be characterized by the gain medium from which the light originates (gas, liquid, solid, or plasma), the wavelength, whether the beam is continuous or pulsed, and, of course, by size and power [Trä12]. In addition, the laser beam can either be strongly focused to achieve a high energy density in one spot, or highly collinear to achieve fairly high energy density at a larger distance.

For weed control, CO_2 (gas) lasers were used first [CG74; Mar+12; Wöl+08]. More recently solid-state lasers have been used, either pure solid state [ASS22; Col+21; Kai+13] or semi-conductor [RA20]. Solid-state lasers require less maintenance and are less susceptible to vibrations than gas lasers, making them attractive for mobile applications.

We selected a solid-state thulium laser (Futonics Laser GmbH, Katlenburg-Lindau, Germany). This type of laser generates light with a wavelength of 2 µm which is very close to the absorption band of liquid water at 1950 nm, meaning that a large fraction of the energy can be expected to be absorbed by water-rich plant tissue.

The laser beam is guided through an optical fiber towards a machining head which contains the optics to create a converging beam with the desired focal distance. To use this beam for cutting it is aimed at a targeting device that uses two actuated mirrors to deflect the beam. The resulting working area measures 200x200 mm on the ground. The laser has an output power of 20 W, a beam diameter of 6 mm, focal distance of 500 mm. The resulting spot size is 0.2 mm. Active water cooling is needed.

The laser apparatus, the machining head, and the cooling system are mounted in a box with approximate dimensions 0.67m x 0.83m x 0.75m (L,W,H) as shown

in Figure F.1b. The distance from the bottom of the box to the ground when the implement is attached to the vehicle is approximately 0.15 m. The box is open at the bottom so that the laser beam can reach the ground surface. The box is fitted with a skirt to close the gap between the box and the ground and to prevent reflected laser light from escaping from the box.

F.3.1.4 Electric weeding

The use of electricity to damage or kill weeds was first proposed more than 100 years ago. In electric weeding, electricity flows through a circuit that starts at an electrode that touches the plant, continues through the leaves, stems, and roots, and is completed through the earthing of the weeding apparatus. Damage results from the heating of the plant due to the application of the electrical current.

Both direct current and alternating current approaches are described in the literature [DB84]. Currently, available commercial apparatus includes RootWave [3] and Zasso [4], both of which use AC (5-15 kV, 50 Hz, 0.5-2.0 A). These machines are used to control weeds in arable cropping and consequently have a large power consumption (up to 30 kW). They are not suitable for controlling single plants or for mounting on a small mobile vehicle.

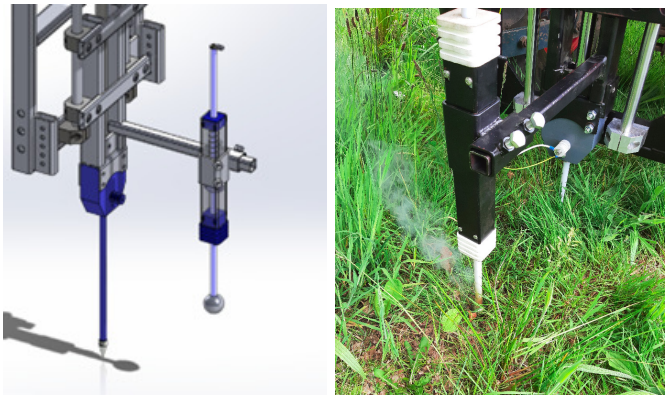
We developed an alternative electric weeding solution based on a 12 kW lamp driver (Nedap N.V., Groenlo, The Netherlands). Device inputs are a 3-phase 360 - 528 Vrms, 48-60 Hz, and max 18 A. Output is controlled within 5 percent accuracy between 2.7 - 11.7kW with an output frequency of 45 kHz, voltage 900 - 1460 Vrms, and current 10 Arms (maximum). The device is controlled through Modbus.

We developed an electric weeding implement that can be attached to the robotic vehicle and can be used for electric weeding in the field. The implement has two electrodes, a positive (control) electrode that is in contact with the *Rumex* plants and a ground electrode that goes into the soil as shown in Figure F.2. The ground electrode has a sharp end to ensure easy entry into the soil. The control electrode has a bulbous end which, when pressed onto the top of the taproot, ensures maximum contact with the *Rumex* plants. Several components were installed in a kilo Volt isolated construction to isolate the high-frequency, high-voltage electricity. The entire assembly is moved vertically using a linear actuator (LA36; LINAK, Breda, The Netherlands). The ground electrode is longer than the control electrode, ensuring that the electrical circuit will be complete when the control electrode touches the plant. The completed weeding implement consists of a camera, computers, and the electric weeding assembly as shown in Figure F.1c.

F.3.1.5 EGNSS Localization

We have developed two solutions for the localization with EGNSS: one high-class solution and one low-class solution. Both variants are integrated into the same custom

[3] https://rootwave.com/
[4] https://zasso.com/



(a) (b)

Figure F.2: The electric weeding implement has two electrodes. The ground electrode has a sharp end to ensure an easy entry into the soil next to the plant. The control electrode has a bulbous end which is expected to be positioned closest to the root at the joint stem position of the *Rumex* plant. Both electrodes are controlled with linear actuators.

IP65 enclosure with the same hardware and software interfaces, allowing for smooth interchangeability on the robot. We use a serial port interface (RS232) as a communication interface to the robot. The same interface is used to receive the correction required for the RTK positioning. Both solutions support Galileo RTK corrections for high-accuracy applications, either by using corrections services through mobile networks or by using another unit of the EGNSS subsystem as a local base station. The latter serves as a fallback solution for remote areas where mobile networks are not accessible.

The high-class solution is based on a Trimble OEM module with dual antenna, including Galileo E5 (AltBOC) and E6 support as key differentiators [5]. Key factors for the selection of the Trimble module for the high-class receiver were the dual antenna support and promised support of forthcoming Galileo signals and features in the form of firmware upgrades.

The low-class solution is based on state-of-the-art multi-frequency ICs from u-blox with E1/E5b support. The receiver incorporates two u-blox Zed F9P chips, while the second chip is required to perform the heading calculation based on the RAW GNSS data. The Zed F9P is the first low-cost multifrequency and multi-constellation chip available in the mass market.

[5] https://www.trimble.com/Precision-GNSS/BD992-Board.aspx

F.3.2 Software to control the robot

The software to control the robot runs on a PC with Ubuntu 18 and is based on ROS [Qui+09]. The PC is part of the implement. The laser is controlled via a Windows DLL, therefore the PC in the laser implement hosts a virtual machine (VM) with Windows; the VM runs a ROS node and accesses the laser DLL. Localization of the robot is realized by fusing the GNSS-derived pose with data from the IMU and the wheel encoders. The extended Kalman filter from the robot_localization package is used for this purpose.

The electric weeding implement follows the same setup. The vertical movement of the electrode assembly is controlled via CANbus. The high-voltage electricity is controlled via Modbus. The PC in the electric weeding implement does not need a Windows virtual machine.

The state machine (see Figure F.3) of the control software has the following states:

- FOLLOW_LINE: In this state, the robot follows a pre-planned back-and-forth route that covers the entire field. Course corrections are based on the perpendicular distance to the line and the difference between robot heading and line heading. While in this state, the weed detection algorithm continuously processes the images captured by the front camera. When a weed is detected, the robot switches to the next state.

- NAVIGATE_TO_RUMEX: In this state, the robot navigates to the detected weed. Course corrections are based on the difference between the robot heading and the heading of the line connecting the robot position and weed position. The robot transitions to the next state when the (laser or electric weeding) implement is positioned directly on top of the weed.

- GENERATE_LASER_TASK: In this state, the robot generates the pattern that the laser will burn (currently: a circle centered on the weed). This takes approximately one second and at the end of it, the robot switches to the next state.

- DESTROY_RUMEX: In this state, the laser burns the pattern that was generated in the previous state. When the pattern is finished, the robot switches to the next state.

- MOVE_BACK_TO_LINE: When the robot enters this state, it has deviated some distance from the pre-planned path to reach the *Rumex* plant that was detected. We have found that often from this position the robot detects a *Rumex* that is yet further away from the pre-planned path, and this process may repeat itself. This results in unpredictable behaviour where the robot wanders over the field and may never return to the original path. To ensure predictable behaviour, in this state, the robot is controlled using course corrections that make it follow the pre-planned route but it does not switch to navigate_to_rumex until it

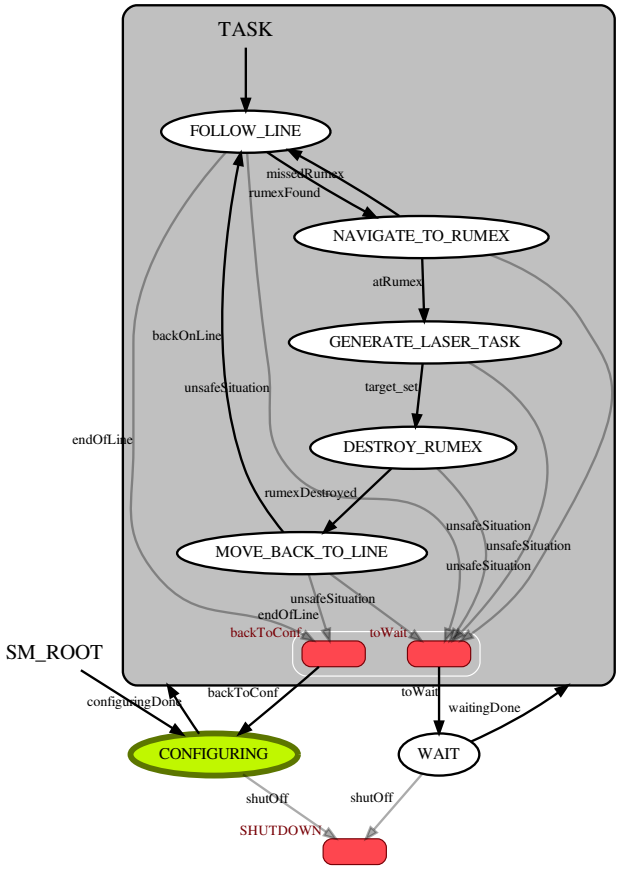


Figure F.3: State machine for the laser implement including five different states: FOLLOW_LINE, NAVIGATION_TO_RUMEX, GENERATE_LASER_TASK, DESTROY_RUMEX, MOVE_BACK_TO_LINE.

comes within a small (preset) distance of the route. At that point, it switches to the FOLLOW_LINE state.

For the electric implement, the laser-specific states (i.e. GENERATE_LASER_TASK and DESTROY_RUMEX) are switched out and replaced with relevant functionalities of the electric weeding device, while the other states remain.

F.4 Experimental evaluation of individual components

F.4.1 Robot navigation

Global navigation was evaluated by monitoring the actual position of the robot while following a pre-planned path as shown in Figure F.4. On the straight lines, the deviations from the path were 0.014 m +- 0.09 m. During the curves, the deviations were 0.038 m +- 0.11 m. Deviations are largest when executing ninety-degree turns which is a direct consequence of the chase the carrot approach.

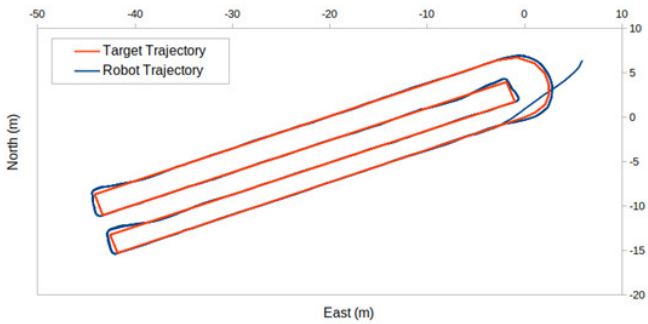


Figure F.4: The pre-planned trajectory (red) vs. the actual robot movement (blue). Deviations are largest on abrupt changes of orientation, such as ninety-degree turns.

F.4.2 Weed detection

For an in-depth analysis of the collected dataset *RumexWeeds* and a corresponding ablation study of our deployed YOLOX model, we refer to our previous work from Güldenring et al. [GEN23]. In the following, we will present a summary.

Dataset *RumexWeeds* The dataset consists of 5,510 first-stage camera images with 15,519 bounding box annotations. It includes image sequences with accompanying navigational sensor data (GPS, IMU, GNSS), which assures flexibility for future model developments – temporal information, as well as navigational information, can be included to improve the model's robustness. The dataset was collected in 2021

at three different farms and 4 different days. We assure to collect the data around 3 weeks after the grass has been cut because the weeds have grown enough to be detected, while the grass remains fairly low. Figure F.5 shows example images from different fields and days.

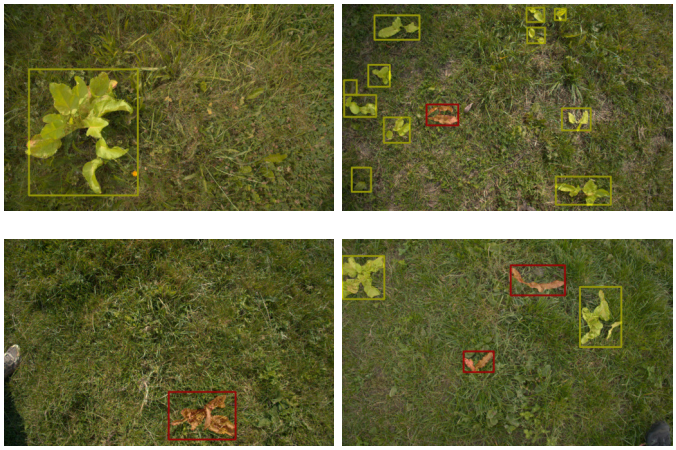


Figure F.5: Sample images from the dataset *RumexWeeds* [GEN23]. Bounding box annotations and corresponding manual pixel-wise annotations are shown. The dataset differentiates between two sub-species (*Rumex obtusifolius L.* in yellow and *Rumex Crispus L.* in orange), however, the deployed model treats them as one class because they are equally undesired.

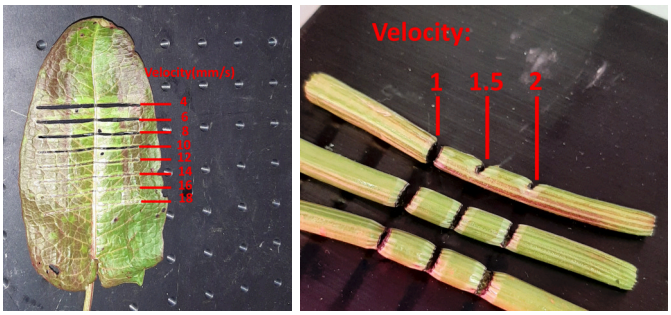
YOLOX model An anchor-free YOLOX model [Ge+21] is deployed to the robot, which was at that point in time the best-performing YOLO variant. Due to modularity, it can be easily updated to a better model. The model takes an image input size of $[640 \times 640]$ and runs in real-time on the Jetson Xavier NX with a FPS of 46.7. It is applied to the first-stage as well as the second-stage image data. On the test split of *RumexWeeds* [GEN23], the model achieves an $mAP_{50:95} = 29.0, mAP_{50} = 55.9$. Furthermore, Güldenring et al. [GEN23] show that the detection performance $mAP_{50:95}, mAP_{50}$ is significantly higher for bigger objects.

Once the robot has detected a plant at the first stage, it drives over the plant and another detection iteration is performed on the RGB image of the second-stage camera to precisely identify the plant location for weed removal while the robot stands still. The plant location serves as input for the weeding unit.

F.4.3 Laser weeding

The scanning speed is an important parameter affecting the success of laser weeding. We determined experimentally that the laser cuts readily through the leaves and the

leaf stems of *Rumex*. Leaves can be cut with a scanning speed of up to 10 mm/s (Figure F.6a) but to cut reliably through some of the thicker leaf stems the speed must not be more than 1 mm/s (Figure F.6b).



(a) **(b)**

Figure F.6: The laser is applied with varying velocities to cut (a) a leaf and (b) thick leaf stems. Our experiments reveal that leaves can be cut with a velocity of up to 10 mm/s. On the contrary, for leaf stems a much lower velocity of $\leq 1mm/s$ is required to perform a thorough cut.

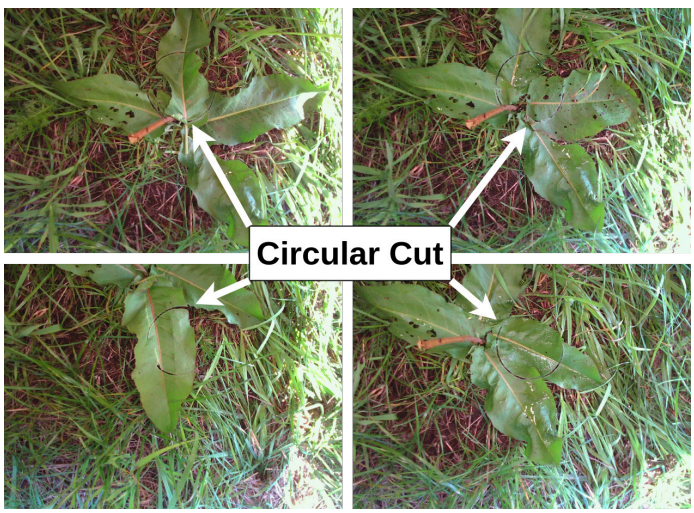


Figure F.7: The laser implement performs a circular cut on four artificial plants, where four leaves are arranged around a stick. The distance of the center of the burned circle from the artificial plant center is between 0.05 and 0.1 cm for three out of four plants.

The effectiveness of applying laser weeding depends to a large extent on the accu-

rate detection of the plant and accurate positioning of the laser implement such that it can damage the plant. We quantified this accuracy in an experiment conducted on 15 September 2023. An artificial *Rumex* plant was made by grouping four *Rumex* leaves around a stick in the ground. The robot started several meters away, detected the plant, positioned itself for lasering, and then burned a circle with a diameter of 0.1 m. The burn circle was centered on the center of the bounding box from the weed detection. After burning, the damaged leaves were replaced. The process was executed four times. For our field experiments, the scanning speed was set to 5 mm/s which is sufficient to burn through leaves and small stems but will not completely cut some of the larger leaf stems. Burning a circle with 0.1 m diameter takes approx. one minute.

Results are shown in Figure F.7. These images were taken by the camera that is located inside the laser implement box, immediately after completion of the burning process. It cannot be determined with certainty where the center of the artificial plants is located. Still, a reasonable estimate is that the distance from the center of the circle to the center of the plants was between 0.05 and 0.10 m in three of the four cases (in the remaining case that distance was approx. 0.2 m). This indicates that a circle with a diameter of 0.2 m would have damaged all leaves in three of four cases.

F.4.4 Electric weeding

F.4.4.1 Laboratory experiments

We conducted several experiments in 2021 and 2022 with potted plants to determine how to damage *Rumex* plants with electricity. These experiments showed that effective treatment could be achieved when plants were in moist soil when one electrode (cable with isolation removed) was stuck into the soil and the other electrode was in contact with the plant, exposed to a current of 2 A for several minutes. Effective treatment could be achieved either by inserting the second electrode into the tap root or by wrapping it tightly around several large leaves.

Based on the results of these initial experiments, a more systematic evaluation of the effectiveness of electric weeding was performed. On 5 September 2022, we collected 24 *Rumex* plants from the field. We measured the weight of each tap root by removing all leaves and washing and drying the tap root. After this, the tap roots were planted in pots and kept in a greenhouse for 24 days, which was a sufficient amount of time to regrow the leaves (Figure F.8).

On 29 September, the number of leaves per plant was counted and plants were randomly assigned either for electric weeding treatment or as control plants. Plants in the weeding treatment were exposed to electric current 2 A for 3 minutes, with one electrode stuck in the soil and the other electrode wrapped tightly around the leaves. After treatment, all plants were returned to the greenhouse. Twenty-one days after the treatment, all plants were harvested and plant fresh weight, root length, root diameter, leaf number, and root damage severity were measured. Various levels of root damage are shown in Figure F.9.

(a) **(b)**

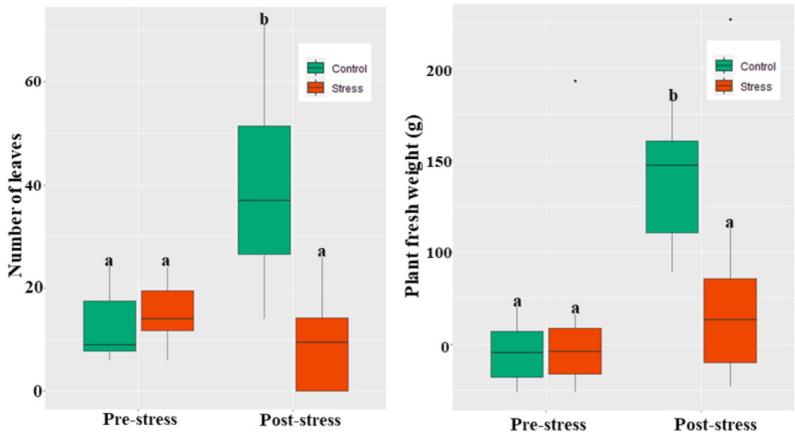
Figure F.8: Experimental setup of *Rumex* plants growing at 20 - 27°C temperature and 273 µmol/m2/s light intensity on (a) day 0 - sample collection: emerging leaves removed and potted in new soil; and (b) day 22 - two days before the application of electric weeding.

The results of this experiment indicate that the electric treatment resulted in severe damage to the roots, a strongly reduced number of leaves, and virtually no increase in biomass after the plants had been treated (Figure F.10 and Figure F.11).

F.4.4.2 Field experiments

The electric weeding implement was tested in the field. On 24 May 2023, the implement was attached to a tractor and powered with mains electricity. The movement of the electrodes was verified by remotely sending commands to the actuator until both electrodes were in the correct position. Electric weeding was also started remotely using a Modbus interface module. Electric weeding time was set to 60 seconds and the process was repeated between 3 and 10 times. After 37 days the treated plants were inspected and it was determined that all six plants had been destroyed.

Figure F.9: The red arrows indicate damage sites on the root. The damage severity of the *Rumex* plants increases from left to right; 0: healthy (undamaged) root and shoot, 1: healthy root with some wilted leaves, 2: $\leq$ 30% damage to the root, 3: 31% - 60% damage to the root, and 4: $>$ 60% damage to root with onset of necrosis.



(a) (b)

Figure F.10: Physiological parameters of control and stressed *Rumex* plants indicating (a) number of leaves, and (b) fresh weight before (pre-stress) and after (post-stress) the electrocution treatment. Here, the number of leaves includes both mature and emerging leaves.

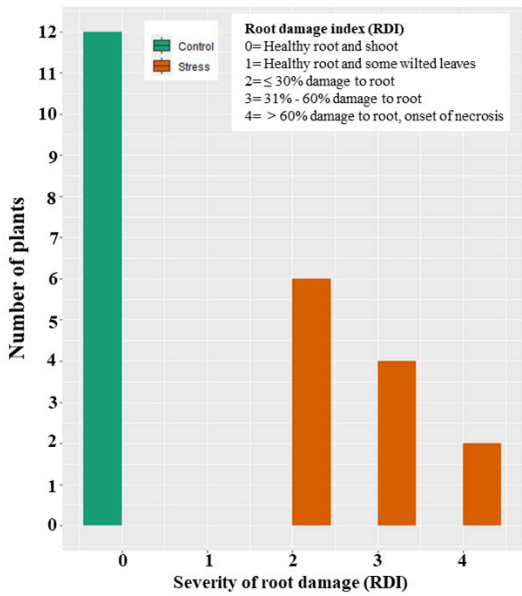


Figure F.11: Distribution of control and stressed *Rumex* plants across the root damage index at the end of experimentation.

F.4.5 EGNSS localization

We compare the high-class and low-class solutions in an outdoor scenario with an open sky to validate the horizontal positioning accuracy in RTK mode. In Table F.1, we show the performance of both receivers against the reference signal, which was obtained in a post-processing step.

	High-class	Low-cost
%fix	97.80%	99.14%
RMSE (m)	0.014	0.023
MAE (m)	0.008	0.004
Max (m)	0.150	0.273

Table F.1: Positioning error of the high-class and the low-cost receivers in RTK mode. It is compared against the reference signal.

The obtained results are within the expected limits for the specs of the high-class and low-cost EGNSS chips, which is 1 cm + 1ppm CEP-50 in ideal conditions. This allows the use of both the low-cost and the high-class solution in the weeding implements. The EGNSS limit allows for an accurate mapping of the weed and navigation towards the weed.

In a second validation step, both solutions were integrated into the robot and tested on an actual *Rumex* field. The electrical and communication interfaces were successfully validated. For both solutions, the position and heading accuracy results were within the limits stated by the EGNSS chip manufacturers, both for RTK mode and single positioning mode.

F.5 Qualitative evaluation of integrated laser weeding implement

Within the scope of the project, we have reached the point to test the laser implement fully integrated with the robot. However, the development of the electric weeding component was delayed in the course of the project, and due to time constraints, we only tested the weeding implement attached to a tractor as described in section F.4.4.2.

For the real-world evaluation, we have deployed the CEOL robot from AGreen-Culture integrated with the laser weeding implement to real grassland fields containing *Rumex obtusifolius L.*. We considered fields in three different countries: France (Cluny, June 2023), Denmark (Taastrup, August 2023), and the Netherlands (Grashoek, June 2023).

In all locations, we initially evaluated individual system components, followed by a full system integration as specified in the following.

- Weed Detection: The robot was positioned in such a way that the first-stage camera was pointing toward weeds. An external monitor was connected to the robot to demonstrate the detection performance of the vision system. On the external monitor, the input image with the corresponding detections – i.e. bounding boxes – was shown.
- Robot Navigation: The robot navigates along a straight line on the field, performs a U-turn at the end of the field, and drives back on the next line. It demonstrated the robot navigation using the EGNSS localization of the implement.
- Full System Integration: The robot starts to navigate along the predefined path. Once a weed is detected, the robot drives over it and removes the weed with the laser unit. Then, the path following is resumed.

The integrated robot was evaluated and demonstrated to an audience of farmers and researchers in all three fields, proving its field-independent functionality. The prototype was able to perform the expected tasks: weed detection, navigation, and weed removal using laser techniques. We refer the reader to our demonstration video, summarizing the performance of the robot and the two proposed implements: https://youtu.be/3LOMdMfIjOs.

F.6 Economic assessment

An economic assessment has been performed to gain insight into the commercial viability of the developed technologies for robotic weed control in grassland fields. The robotic solution could be bought by farmers in the same way as other equipment is bought or, alternatively, it could be provided in the form of robotics-as-a-service (RAAS). In the former case, there is a direct investment with capital expenses, in the latter case there is no investment by the farmer but there are operational expenses.

In the assessment the options of laser weeding and electric weeding are taken into account, using two implement options to be attached to one generic robot platform. A market review showed that for the European Union, there is a target of 25 percent organic land use for farming, however, that target can hardly be reached without additional technological innovations for weed control in grasslands, a positive outcome for our developments. Farmer interviews provided us with insights that ”pay per weed” is also a viable option, as five to ten Euro per removed weed plant would be accepted based on the amount of work manual removal would take as well.

In Galirumi we calculated the manufacturing costs for producing five robots for electric weeding at 70 kEuro per robot, and five robots for laser weeding at 72.5 kEuro. These manufacturing costs were validated by our commercial and industrial partners in the project. When we increase the production numbers to 25 units, the manufacturing costs decrease to 51.3 kEuro for both weeding technologies. Given that the market is urgently asking for non-chemical weed control options, it is expected

that the break-even number of machines to be sold will be reached after 3 years, with a total of 33 machines to be sold by the manufacturer.

However, bringing a robotic solution into the market does not end with production costs being at break-even for the manufacturer. Sales support, updates, and servicing of the robots also need to be taken into account. Based on the experiences of the business developers involved in Galirumi, the margin for the sales price was determined and the sales price was estimated at a total of 95 kEuro for the robotic weeding solution. This will enable a healthy business model for the manufacturer as well as the support organization. Even at the low sales numbers that are expected in the first years after market introduction, the business case is valid, having a relative margin of close to 20 percent which is required to run a business in agriculture and is needed to get funding from investors or banks.

F.7 Discussion

We have designed and constructed two implements that are meant to be attached to a robotic vehicle and can then be used to control *Rumex*. The resulting weed control robot was subjected to initial testing in realistic field conditions and was found to work as expected.

The robot meets most of the requirements that were elicited in conversations with stakeholders. The robot is able to work while it is raining and it can be transported in a van or on a small trailer. The electric-vehicle requirement is partly met. We used a vehicle with an electric drive train, whose battery is charged by a generator that is powered by a diesel engine. This offers the advantage of never running out of power, even during long days of testing. A fully electrical version of the vehicle is available. The robot is currently not able to weed in darkness but the addition of some lights would make working at night possible.

There are some limitations to our results that are associated with the conditions under which the work was carried out.

Navigation, such as line-following and maneuvering to approach a detected weed, worked well in diverse environments in France, Denmark, and The Netherlands but we did not encounter the full range of soil types (sand, clay, peat) and soil conditions (dry, wet) in these countries. Weed detection worked very well overall but the algorithm tends to classify broad-leaved weeds such as dandelion (*Taraxacum officinale* L.) and broad-leaf plantain (*Plantago major* L.) as *Rumex*. This is not a serious limitation, as many pastures contain mostly rumex, and dandelion is considered a weed as well. Nevertheless, the selectivity of the algorithm could be improved by collecting a more diverse data set, also from fields with other plant compositions than just *Rumex* and grass, and re-training the weed detection algorithm. Now that the robot is operational, it can be used to collect new image data. Annotating these images is expensive. A potential solution to improve the efficiency of annotation is active learning, a process whereby only those images are annotated where the algorithm is uncertain about

the classification (e.g. Blok et al. [Blo+22]). Other approaches include unsupervised pre-training to generate domain-specific weights and the generation of synthetic data.

The selected thulium laser cuts readily through leaves and leaf stems. We established a laser burning speed that works in most conditions but we did not systematically test how the maximum effective burning speed is affected by leaf age, thickness, and hydration status, and by the thickness of leaf stems.

Electric weeding was developed using a high-voltage source from a different domain. In the course of our work, this source was adapted to the needs of the project. In the initial stages, it was not possible to limit current and voltage as intended which limited our ability to investigate in a systematic manner the effect of voltage and current on the effectiveness of treating *Rumex* with electricity. The electric implement has so far only been tested when attached to a tractor (using the same three-point linkage that is on the robot).

Rumex is the predominant weed in temperate grassland and *Rumex* control has received much attention. This allows us to compare our results to results reported in the literature, both in terms of the complete robot and in terms of components of *Rumex* detection and *Rumex* control.

In earlier work, we described a robot to detect and control *Rumex* [Eve+11]. The current robot is an improvement in all aspects, with the separation of vehicle and implements for lower cost, with ROS-based software that is easier to maintain than purposely written software, with deep-learning based weed detection that functions well under all kinds of illumination (including heavy shadows), and with better weed control methods.

The vision-based detection of *Rumex obtusifolius L.* (broad-leaved dock) has been explored by various authors since the 2000s [Pol+07; Dür+08; BF17]. In recent years, the usage of deep learning techniques has become a popular choice because it allows one to develop models performing well in varying field conditions, such as illumination changes, different weather conditions, and varying growth stages. The detection robustness of those models increases with the quality of the underlying data containing the above-mentioned variations. While most deep learning models for the detection of *Rumex* [KTN19; SAŠ19; Val+22] are trained on a small set of annotated images, we took the effort to collect a larger dataset on different days and locations throughout the year. Furthermore, we apply a deep learning architecture that allows for classification and localization (i.e. bounding boxes) in one inference step in real-time on the robot.

The use of a laser to control weeds has been described by several authors. Typically the laser is used to damage the meristem (growing point) of small seedlings. However, *Rumex* has many meristems that are buried just below the soil surface and therefore cannot be damaged with a laser. Therefore, in a novel approach, we use the laser as a cutting tool to damage entire leaves. This requires more time and more (electric) energy than only damaging a meristem. Fortunately, the number of Rumex plants to be destroyed in dairy farming is far smaller than the number of seedling weeds in arable farming.

Electric weeding has also received much attention in the literature. Experiments using single plants to determine which electric treatment (AC or DC, voltage, current, duration) is most effective have been described. Currently, available equipment for field use is focusing on large-area applications. Little information is available on controlling single plants recent exceptions [Leh+22]. Our approach to controlling single *Rumex* plants with electricity is novel.

Chemical control of *Rumex* is unattractive when the whole pasture must be sprayed. Several camera-based selective spraying solutions have recently reached the market, including Ara [6], AiCPlus [7], and the RXF600 system [8]. These systems allow a very large reduction in chemical use relative to whole-field spraying. But they are still chemical and therefore not helpful to organic farmers.

The initial results presented here are a good starting point for further work. So far we have assumed the position of the center of the *Rumex* rosette to be the same as the center of the YOLO bounding box. This is not always accurate because of the asymmetry of the plant. For a more accurate center position, a multi-task model to predict both the enclosing bounding box as well as the joint-stem position can be considered in the future [LGN23].

Laser burning currently takes more time than we had anticipated. We could use a more powerful laser and reduce cutting time. It would be sensible to add sensors to detect heat and prevent fire. We have observed that the laser sometimes sets the grass on fire. However, this happened only in poorly maintained grassland where grass residue from a previous cut was left in the field and had become completely dry; this is not likely to happen in normal production dairy pasture in the Netherlands where all grass is picked up. Alternatively, we could aim for the generation of more efficient laser patterns. The second-stage camera of the current robot provides high-resolution images that can be used to perform a fine-grained analysis of the plant by locating the individual veins and stems of the leaves [GAN24]. From this information, laser patterns consisting of point and line cuts could be generated to reduce the overall cutting time. Additionally, the laser cutting speed can be dynamic, depending on whether a stem or leaf cut is performed.

Our work compares two contrasting weed control methods, both of which have specific advantages and disadvantages. An advantage of laser weeding is that no contact with either soil or weed is needed, thus there is little wear on the implement (which is fragile in comparison to conventional agricultural implements). The laser is heavy (approx. 20 kg) and the water-cooling increases that weight. However, it would be possible to use a lighter laser. That would increase the time required to damage a plant, but, importantly, it would allow all components to be lighter as well. It may be possible to reduce weight and size of a laser weeding robot to the point where its power requirements can be met by solar panels. A disadvantage of laser weeding is that the treatment must be done at least several times per year. A robot that uses

[6] https://ecorobotix.com/en/ara/
[7] https://www.agrifac.com/nl/optionele-spuittechniek/aicplus/
[8] https://rumex-gmbh.de/

electric weeding has the advantage that it can kill the weed in a single treatment but it does not seem likely that it can be powered by solar panels.

F.8 Conclusion and future work

We have constructed a robot to control *Rumex*. The robot consists of a vehicle that can be connected to two different implements. The laser weeding implement has been evaluated in practice and functions well. The electric weeding implement needs further work.

The current robotic weeding system can be considered a prototype. Work toward practicality is needed, but already the eventual product seems to be commercially viable.

An important next step will be to quantify the effect of using the weeding robot on the abundance of *Rumex* in real farming conditions. To this end, we aim to conduct an experiment over an entire growing season in which part of a commercial pasture is left untreated while another part is treated with the robot. The effect of using the robot will be determined by measuring the number of *Rumex* plants and their weights at the end of the growing season in both parts of the pasture.

F.9 Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project "Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)", H2020-SPACE-EGNSS-2019-870258.

References

[ASS22] Christian Andreasen, Karsten Scholle, and Mahin Saberi. "Laser Weeding With Small Autonomous Vehicles: Friends or Foes?" In: *Frontiers in Agronomy* 4 (2022). ISSN: 2673-3218. DOI: 10.3389/fagro.2022.841086.

[BF17] A. Binch and C.W. Fox. "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland." In: *Computers and Electronics in Agriculture* 140 (August 2017), pages 123–138.

[Bir+20] Simon Birrell et al. "A field-tested robotic harvesting system for iceberg lettuce." In: *Journal of Field Robotics* 37.2 (2020), pages 225–245. DOI: https://doi-org.proxy.findit.cvt.dk/10.1002/rob.21888.

[Ble+16] Timo Blender et al. "Managing a Mobile Agricultural Robot Swarm for a seeding task." In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pages 6879–6886. DOI: 10.1109/IECON.2016.7793638.

[Blo+22] Pieter M. Blok et al. "Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes." In: *Computers and Electronics in Agriculture* 197 (2022), page 106917. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.106917.

[CG74] Richard Couch and Edward O. Gangstad. "Response of Waterhyacinth to Laser Radiation." In: *Weed Science* 22.5 (1974), pages 450–453. DOI: 10.1017/S0043174500037991.

[Col+19] Guy R. Y. Coleman et al. "Using energy requirements to compare the suitability of alternative methods for broadcast and site-specific weed control." In: *Weed Technology* 33.4 (2019), pages 633–650. DOI: 10.1017/wet.2019.32.

[Col+21] Guy Coleman et al. "Low Energy Laser Treatments Control Annual Ryegrass (Lolium rigidum)." In: *Frontiers in Agronomy* 2 (2021). ISSN: 2673-3218. DOI: 10.3389/fagro.2020.601542.

[Cub+20] Sergio Cubero et al. "RobHortic: A Field Robot to Detect Pests and Diseases in Horticultural Crops by Proximal Sensing." In: *Agriculture* 10.7 (2020). ISSN: 2077-0472. DOI: 10.3390/agriculture10070276.

[DB84] M.F. Diprose and F.A. Benson. "Electrical methods of killing plants." In: *Journal of Agricultural Engineering Research* 30 (1984), pages 197–209. ISSN: 0021-8634. DOI: https://doi.org/10.1016/S0021-8634(84)80021-9.

[DIP+85] M. F. DIPROSE et al. "Use of electricity to control bolters in sugar beet (Beta vulgaris L.): a comparison of the electrothermal with chemical and mechanical cutting methods." In: *Weed Research* 25.1 (1985), pages 53–60. DOI: https://doi.org/10.1111/j.1365-3180.1985.tb00617.x.

[Dür+08] Lorenz Dürr et al. "Machine vision detection and microwave-based elimination of rumex obtusifolius L. on grassland." In: 2008.

[Eve+09] F K van Evert et al. "Real-time vision-based detection of Rumex obtusifolius in grassland." In: *Weed Research* 49.2 (2009), pages 164–174.

[Eve+11] Frits K. van Evert et al. "A robot to detect and control broad-leaved dock (Rumex obtusifolius L.) in grassland." In: *Journal of Field Robotics* 28.2 (2011), pages 264–277.

[Eve+20] F K van Evert et al. "Weekly defoliation controls, but does not kill broad-leaved dock (Rumex obtusifolius)." In: *Weed Research* 60.2 (2020), pages 161–170. DOI: https://doi.org/10.1111/wre.12407.

[GAN24] Ronja Güldenring, Rasmus Eckholdt Andersen, and Lazaros Nalpantidis. "Zoom in on the Plant: Fine-Grained Analysis of Leaf, Stem, and Vein Instances." In: *IEEE Robotics and Automation Letters* 9.2 (2024), pages 1588–1595. DOI: 10.1109/LRA.2023.3346807.

[Ge+21] Zheng Ge et al. "YOLOX: Exceeding YOLO Series in 2021." In: *arXiv preprint arXiv:2107.08430* (2021).

[GEN23] Ronja Güldenring, Frits K. van Evert, and Lazaros Nalpantidis. "RumexWeeds: A grassland dataset for agricultural robotics." In: *Journal of Field Robotics* (2023). DOI: https://doi.org/10.1002/rob.22196.

[Gül+21] Ronja Güldenring et al. "Few-leaf Learning: Weed Segmentation in Grasslands." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[Hei+01] T Heisel et al. "Cutting weeds with a CO2 laser." In: *Weed Research* 41.1 (2001), pages 19–29. DOI: https://doi.org/10.1046/j.1365-3180.2001.00212.x.

[HIR+12] S HIREMATH et al. "The role of textures to improve the detection accuracy of Rumex obtusifolius in robotic systems." In: *Weed Research* 52.5 (2012), pages 430–440. DOI: https://doi.org/10.1111/j.1365-3180.2012.00931.x.

[Kai+13] Stefan Kaierle et al. "Find and Irradiate — Lasers Used for Weed Control." In: *Laser Technik Journal* 10.3 (2013), pages 44–47. DOI: https://doi.org/10.1002/latj.201390038.

[KTN18] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Image-Based Recognition Framework for Robotic Weed Control Systems." In: *Multimedia Tools Appl.* 77.8 (April 2018), pages 9567–9594. ISSN: 1380-7501.

[KTN19] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Deep learning-based visual recognition of rumex for robotic precision farming." In: *Computers and Electronics in Agriculture* 165 (2019), page 104973. ISSN: 0168-1699.

[Lat+07] Roy Latsch et al. "Control of Rumex Obtusifolius L . in Grassland Using Microwave Technology." In: 2007.

[Lat+17] R Latsch et al. "Controlling Rumex obtusifolius by means of hot water." In: *Weed Research* 57.1 (2017), pages 16–24. DOI: https://doi.org/10.1111/wre.12233.

[Leh+22] Erik A. Lehnhoff et al. "Electricity as an effective weed control tool in non-crop areas." In: *Weed Research* 62.2 (2022), pages 149–159. DOI: https://doi.org/10.1111/wre.12523.

[LGN23] Jiahao Li, Ronja Güldenring, and Lazaros Nalpantidis. "Real-Time Joint-Stem Prediction for Agricultural Robots in Grasslands Using Multi-Task Learning." In: *Agronomy* 13.9 (2023). ISSN: 2073-4395. DOI: 10.3390/agronomy13092365.

[LS14] Roy Latsch and Joachim Sauter. "Optimisation of hot-water application technology for the control of broad-leaved dock (Rumex obtusifolius)." In: *Journal of Agricultural Engineering* 45 (December 2014), page 137. DOI: 10.4081/jae.2014.239.

[Mar+12] Christian Marx et al. "Design and application of a weed damage model for laser-based weed control." In: *Biosystems Engineering* 113.2 (2012), pages 148–157. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2012.07.002.

[Ols+19] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning." In: *Scientific Reports* 9.2058 (1 February 2019). DOI: 10.1038/s41598-018-38343-3.

[Pal23] Paltech. *Paltech*. https://paltech.eu/en/. Accessed: 2024-01-29. 2023.

[Par+] Soran Parsa et al. "Modular autonomous strawberry picking robotic system." In: *Journal of Field Robotics* n/a.n/a (). DOI: https://doi-org.proxy.findit.cvt.dk/10.1002/rob.22229.

[Pol+07] G. Polder et al. "Weed Detection Using Textural Image Analysis." English. In: *EFITA/WCCA Conference*. 2007.

[Qui+09] Morgan Quigley et al. "ROS: an open-source Robot Operating System." In: *ICRA Workshop on Open Source Software*. 2009.

[RA20] Ildar Rakhmatulin and Christian Andreasen. "A Concept of a Compact and Inexpensive Device for Controlling Weeds with Laser Beams." In: *Agronomy* 10.10 (2020). ISSN: 2073-4395. DOI: 10.3390/agronomy10101616.

[RK07] A M RASK and P KRISTOFFERSEN. "A review of non-chemical weed control on hard surfaces." In: *Weed Research* 47.5 (2007), pages 370–380. DOI: https://doi.org/10.1111/j.1365-3180.2007.00579.x.

[SAŠ19] Damian Schori, Thomas Anken, and Dejan Šeatović. "Using Fully Convolutional Networks for Rumex Obtusifolius Segmentation, a Preliminary Report." In: *2019 International Symposium ELMAR*. 2019, pages 119–122.

[ŠKA10] Dejan Šeatović, Hansjörg Kutterer, and Thomas Anken. "Automatic weed detection and treatment in grasslands." In: *Proceedings ELMAR-2010*. 2010, pages 65–68.

[Trä12] Frank Träger. *Springer handbook of lasers and optics*. Volume 2. Springer, 2012.

[Val+22] João Valente et al. "Mapping of Rumex obtusifolius in nature conservation areas using very high resolution UAV imagery and deep learning." In: *International Journal of Applied Earth Observation and Geoinformation* 112 (2022), page 102864. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2022.102864.

[VB01] Clément Vigneault and Diane L. Benoît. "Electrical Weed Control: Theory and Applications." In: *Physical Control Methods in Plant Protection*. Edited by Charles Vincent, B. Panneton, and F. Fleurat-Lessard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pages 174–188. ISBN: 978-3-662-04584-8. DOI: 10.1007/978-3-662-04584-8_12.

[Wey+22] J. Weyler et al. "Joint Plant and Leaf Instance Segmentation on Field-Scale UAV Imagery." In: 7.2 (2022), pages 3787–3794. ISSN: 377-3766. DOI: 10.1109/LRA.2022.3147462.

[Wöl+08] C. Wöltjen et al. "Plant growth depression by selective irradiation of the meristem with CO2 and diode lasers." In: *Biosystems Engineering* 101.3 (2008), pages 316–324. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2008.08.006.