Inaugural-Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Plant Classification Systems for Agricultural Robots

von

Philipp Lottes

aus
Oberhausen, Deutschland

Bonn 2021

Referent:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

1. Korreferent:
Prof. Dr. Cristopher McCool, University of Bonn, Germany

2. Korreferent:
Prof. Dr. Achim Walter, Eidgenössische Technische Hochschule Zürich

Tag der mündlichen Prüfung: 15.01.2021

Erscheinungsjahr: 2021

Angefertigt mit Genehmigung der Landwirtschaftlichen Fakultät der Universität Bonn

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenenen Hilfsmittel angefertigt habe; die aus fremdem Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum	(Unterschrift)

Zusammenfassung

Durch die steigende Weltbevölkerung wächst der Bedarf an Nahrung und Energie kontinuierlich. Als eine der zentralen Nahrungs- und Energiequellen ist die Pflanzenproduktion daher angehalten, höhere Erträge zu erzielen. Unkrautbekämpfung und Düngung sind dabei zentrale Maßnahmen, um hohe Ertragsraten zu erzielen. Heutzutage werden Dünger und Herbizide gleichmäßig und in großen Mengen in den Feldern appliziert. Jedoch müssen wir eine unnötige Verwendung dieser Chemikalien vermeiden, um unsere Umwelt für die künftigen Generationen besser zu schützen.

Autonome landwirtschaftliche Feldroboter bieten das Potenzial für eine drastische Reduzierung der eingesetzten Chemikalien. Roboter können für eine selektive Behandlung einzelner Pflanzen und Unkräuter mit verschiedenen Aktuatoren, wie zum Beispiel selektiven Sprühern, mechanischen Werkzeugen oder sogar Lasern, ausgestattet werden. Die Voraussetzung für die selektive und pflanzenbezogene Behandlung ist, dass die Roboter zunächst die einzelnen Pflanzen und Unkräuter auf dem Feld unterscheiden und lokalisieren können. Mit diesen Informationen können Roboter dann entscheiden, wo und wann die Aktuatorik für eine selektiv Behandlung ausgelöst werden muss. Zudem können unbemannte Flugroboter (UAVs) Felder in großem Maßstab vermessen, ohne dabei mit dem Ackerboden zu interagieren. In Kombination mit einem Pflanzenklassifikationssystem bieten UAVs daher die Möglichkeit, gesamte Bestände in vergleichsweise geringer Zeit zu analysieren.

In dieser Arbeit stellen wir neuartige Pflanzenklassifikationssysteme vor, die Feldrobotern eine selektive Pflanzenbearbeitung direkt während der Überfahrt ermöglichen und Flugroboter in die Lage versetzen, eine Analyse der Bestände aus der Luft durchzuführen. Wir untersuchen traditionelle und modernere maschinelle Lernansätze für die Durchführung der dafür notwendigen Klassifikation der Pflanzen und Unkräuter. Unsere Ansätze basieren entweder auf Random Forests oder auf Fully Convolutional Neural Networks.

Wir schlagen in dieser Arbeit ein bildbasiertes System für die gemeisame Klassifikations von Pflanzen- und Pflanzenstämmen vor. Unser Ansatz erkennt gleichzeitig die Nutz-pflanzen und Unkräuter sowie deren genaue Stammposition. Unser System differenziert zudem die Unkräuter in die Klassen Gräser und

Kräuter. Auf Basis solch automatisch gewonnener Klassifikationsergebnisse, kann ein Roboter die effektivste Behandlung für die aktuelle Situation auf dem Feld auswählen.

Eine große Herausforderung für bildbasierte Klassifikationssysteme besteht darin, dass Feldroboter auf verschiedenen Feldern operieren, und daher oft mit drastischen Veränderungen des visuellen Erscheinungsbildes der Pflanzen, der Unkräuter und des Ackerbodens konfrontiert sind. Daher müssen Pflanzenklassifikationssysteme nicht nur eine hohe Leistung auf bereits bekannten Feldern erbringen, sondern auch robust gegenüber neuen und sich ändernden Bedingungen auf dem Feld sein. Daher zielt diese Arbeit darauf ab, die Generalisierungsfähigkeit von Pflanzenklassifikationssystemen für Roboter, die unter verschiedenen Feldbedingungen arbeiten, zu verbessern. Wir schlagen zwei Klassifikationssysteme vor, die zusätzlich zu den visuellen Informationen aus den Bildern auch die räumliche Verteilung der Pflanzen bei Reihenkulturen ausnutzen können. Diese geometrischen Informationen sind typischerweise innerhalb eines Feldes sowie über verschiedene Felder hinweg ähnlich und daher weniger abhängig vom visuellen Erscheinungsbild der Pflanzen. Wir zeigen, dass unsere Ansätze, welche die räumliche Verteilung der Pflanzen ausnutzen, eine bessere Generalisierungsfähigkeit bei sich ändernden Feldbedingungen bieten als Klassifikatoren, welche ausschließlich visuelle Informationen bearbeiten.

Eine weitere Herausforderung für eine skalierbare Entwicklung robuster Pflanzenklassifikationssysteme ist der Bedarf an vielen und vielfältigen Trainingsdaten. Für eine gute Leistung unter neuen Feldbedingungen muss ein Klassifikator typischerweise zunächst mit zusätzlichen Trainingsdaten angepasst werden. Dabei müssen die Trainingsdaten die Bedingungen des neuen Feldes repräsentieren. Dieses Vorgehensweise benötigt jedoch kontinuierlich neue Trainingsdaten und geht daher direkt mit stetigem Annotationsaufwand einher. Wir stellen einen semiüberwachten Ansatz vor, welcher seine Modelle während der Klassifikationsphase an die aktuelle Situation anpasst. Unser Ansatz kombiniert eine visuelle Klassifikation mit einer geometrischen Klassifikation, welche die relative Pflanzenanordnung ausnutzt. Wir zeigen, dass unser Ansatz mit einem nur einminütigem Annotationsaufwand eine Klassifikationsleistung auf dem gleichen Niveau wie klassisch angepasste Klassifikatoren bietet.

Wir präsentieren desweiteren eine umfassende experimentelle Evaluierung der Klassifikationssysteme unter realen Bedingungen. Dazu haben wir eine große und vielfältige Datenbasis auf verschiedenen Feldern in Mitteleuropa gesammelt und insgesamt dafür etwa 26.500 Bilder händisch annotiert. Die Bilder wurden von verschiedenen Feld- und Flugrobotern und unter verschiedenen Bedingungen aufgenommen. Mit Hilfe unserer Datenbasis evaluieren wir verschiedene Aspekte der Pflanzenklassifikatoren unter Berücksichtigung ihrer Leistung, ihrer General-

isierungsfähigkeit, des erforderlichen Annotationsaufwands und der Nützlichkeit zusätzlicher Nah-Infrarot-Bilder. Zudem vergleichen wir explizit die Performanz der Random Forest Ansätze mit der Performanz der Fully Convolutional Neural Network Ansätze. Unsere Experimente zeigen, dass Fully Convolutional Neural Network Ansätze für die Pflanzenklassifikation generell gut geeignet sind. Sie bieten eine bessere Leistung als Random Forests, sind robuster gegenüber wechselnden Feldbedingungen und liefern ihre Ergebnisse schneller, da Neural Networks dedizierte Hardwarekomponenten ausnutzen.

Alle in dieser Arbeit vorgestellten Pflanzenklassifikationssysteme wurden in mehreren Konferenzbeiträgen und Zeitschriftenartikeln veröffentlicht. Eines unserer UAV-basierten Klassifikationssysteme wurde auf der International Conference on Robotics and Automation (ICRA) als bestes Automatisierungspapier ausgezeichnet und unser semiüberwachter Ansatz wurde auf der International Conference on Robots and Systems (IROS) als Finalist für den Preis für das beste Anwendungspapier ausgezeichnet.

Abstract

DUE to a continually growing world population, the demand for food and energy increases continuously. As a central source of food, feed, and energy, crop production is therefore called upon to produce higher yields. To achieve high crop yields, weed control, fertilization, and disease control are essential tasks. Nowadays, these tasks are performed by uniformly applying large amounts of agrochemicals, such as herbicides and fertilizers, to our fields. At the same time, we need to reduce the ecological footprint of agricultural production to achieve the required sustainability to protect our environment for future generations.

Autonomous agricultural field robots offer the potential for a drastic reduction of applied agrochemicals through selectively treating individual plants and weeds in the field. For selective weeding or fertilizing, a robot can be equipped with different actuators such as selective sprayers, mechanical tools, or even lasers. A prerequisite for selective and plant-specific treatment is that the robots can distinguish and locate the plants and weeds in the field. With this information, the robots can decide where and when to trigger the actuators to perform the treatment selectively. In contrast to ground robots, unmanned aerial vehicles (UAVs) can monitor farmland on a larger scale without interacting with the soil. In combination with a vision-based system for the classification of plants, UAVs serve excellent capabilities to retrieve the status of a field on a per-plant basis in small amounts of time.

In this thesis, we develop novel vision-based plant classification systems that enable agricultural ground robots for online in-field interventions and for aerial robots to perform accurate monitoring of the plantation. We investigate traditional and more modern machine-learning approaches based on random forests and fully convolutional neural networks to perform the necessary classification of the crop plants and weeds. We propose a coupled plant and stem classification system that jointly classifies the crop plants and weeds, further distinguishing herbs from grasses, and additionally provides the precise stem locations at the same time. Based on the classification output, the robot can select the most effective treatment for the current situation in the field.

A major challenge for vision-based classification systems is that agricultural robots need to operate in different field environments under drastic changes in the visual appearance of the plants, weeds, and soil. For real-world applications, plant classifications systems not only need to provide a high performance in known fields but also need to be robust to new and changing field conditions. This thesis aims at improving the generalization performance of plant classification systems for robots that operate under different environmental conditions. We propose two vision-based classification systems that, in addition to visual information, exploit the spatial arrangement of the plants in the case of row crops. Such geometric information is typically similar within and across fields and thus less dependent on the visual appearance of the plants. Our approaches exploiting the spatial arrangement of plants provide superior generalization capabilities to changing field conditions compared to state-of-the-art vision-based classifiers.

A further challenge for scalable development of robust plant classification systems is their requirement for large and diverse training datasets. Typically, a classifier needs to be adapted with additional labeled data representing the conditions of the new field environment. However, this procedure comes at the cost of a continuous effort to label new data. We present a semi-supervised online learning approach that combines purely visual classification with geometric classification exploiting the plant arrangement. We show that with only a one-minute labeling effort, our approach provides a classification performance on the same level as classically re-trained classifiers.

We conduct a comprehensive experimental evaluation of the classification systems under real-world conditions using a wide range of field datasets. We collected a large and diverse database in various field environments located in central Europe consisting of around 26,500 labeled images acquired by different field and aerial robots. Using our database, we evaluate different aspects of the plant classifiers considering their performance, generalization capabilities, needed labeling effort, exploitation of additional near-infrared information, and explicitly compare the random forest performance with the one obtained by fully convolutional neural networks. Our experiments suggest that fully convolutional neural networks are well suited for the plant classification task. They provide a better performance than random forest-based approaches, are more robust to changing field conditions, and provide results faster by exploiting dedicated hardware.

All plant classification systems presented in this thesis have been published in peer-reviewed conference papers and journal articles. One of our UAV-based plant classification systems won the best automation paper award at the International Conference on Robotics and Automation (ICRA). Our semi-supervised online-learning approach for plant classification was a finalist for the best application paper award at the Conference on Robots and Systems (IROS).

Acknowledgements

In the past five years, I have been able to learn a lot of new and exciting things. During my doctoral studies, I met many excellent people who supported me on my way to this work. It was a pleasure for me to work in our Photogrammetry and Robotics Lab in Bonn.

First of all, I would like to thank Cyrill Stachniss for being a remarkable supervisor for me. Already during my master's studies, I noticed the unique working atmosphere in his courses. During my time as a Ph.D. student, I was able to be part of his team and learned a lot about professional work, personal interaction, and leadership in a great team. He always supported me in all my concerns and showed me through his example that a positive attitude is key to master any challenge.

I want to thank Achim Walter for reviewing this thesis. I enjoyed both our fruitful discussions and our collaboration during the Flourish project. I would also want to thank Christopher McCool for reviewing this thesis. We met at my first ICRA conference in 2016 and from those days on, we had fruitful discussions about challenges in the context of modern machine vision in agricultural settings, which helped me to gain more understanding of the technology.

I want to extend my gratitude to all the people in the lab. I enjoyed the company of Igor Bogoslavskyi, Olga Vysotska, Kaihong Huang, Jens Behley, Nived Chebrolu, Xieyuanli Chen, Federico Magistri, Andres Milioto, Lorenzo Nardi, Argentina Ortega Sainz, Emanuele Palazzolo, Timo Röhling, Robert Schirmer, Christoph Siendentop, Ignacio Vizzo, Jan Weyler, Louis Wiesmann, Julio Pastrana, Ribana Roscher, Johannes Schneider, Susanne Wenzel, and Wolfgang Förstner. The many discussions, the joint lunch breaks, and special events have enriched the working time so that I had the feeling of being together with friends. Thank you all!

Special thanks to Nived Chebrolu and Jens Behley for the countless conversations about almost all topics of life. Moreover, our discussions have continuously improved the quality of research in this work. Thank you, even though there was always more work on the plate after almost every discussion.

Special thanks go to Birgit Klein, who helped me with all administrative matters whenever I needed it, and to Thomas Läbe for all the informative conversations about network administration and in-depth technical discussions.

Special thanks go to Jan Weyler, Dario Gogoll, Ferdinand Langer, Jan Quakernack, Jonas Hüssen, Martin Obersheimer, Andreas Kräußling, Jana Kirchdorf and Florian Görlich, Jens Behley, Nived Chebrolu and everyone involved in the extensive work for data labeling. Without your dedication, the studies in this work would not have been possible with this quality, thank you!

My thanks to Diana Becirevic, Lasse Klingbeil, Holger Milz, and Michael Pesch for the collaboration on hardware problems and data gathering at Campus Klein Altendorf.

My thanks to Ralf Pude and his team from the Klein Altendorf campus for their great support of our data collection campaigns over the past five years.

My thanks to the team from ETH Zurich around Achim Walter, Roland Siegwart, Raghav Khanna, Frank Liebisch, and Johannes Pfeifer for numerous joint data collections and papers.

My thanks to the entire Flourish team for a great time on an exciting project. Special thanks to all project members for the unforgettable and formative integration weeks during the project.

My thanks go to Slawomir Sander and the team from Deepfield Robotics, especially Markus Höferlin. Thanks to our cooperation at the beginning of this work, I got an excellent introduction to the topic of agricultural robotics and was able to start my Ph.D. thesis directly from state of the art.

Last but not least, I want to thank my family, who have always supported me. Barbara, Udo, Melanie, Achim and Peter, thank you very much for your tireless support. My special thanks go to Stephanie, who accompanied me on this adventure and was unconditionally at my side every second, and Leo, who gave me motivation even in challenging phases.

The work presented in this thesis is partially supported by the European Commission through the Flourish project, H2020-ICT-644227-Flourish, and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070-390732324-PhenoRob.
The financial support of the EC through the Flourish project and of the DFG through the PhenoRob Cluster of Excellence is gratefully acknowledged.

Contents

Chapter 1

Introduction

A central societal challenge is to meet the increasing demand for food and energy induced by an ever-growing world population [40]. According to Tilman *et al.* [145], we have to double the global yields obtained by crop production by 2050 to meet the forecasted demands. Crop production is the key to satisfy these needs. At the same time, arable land is limited, and the environmental footprint of agricultural production needs to be reduced in order to achieve the required sustainability to protect our environment for future generations. Thus, new approaches are needed for sustainable crop production, and this thesis aims at making a contribution that addresses these challenges.

Plants compete with weeds for the nutrients and water in the soil. For high crop production, weed control, fertilization, and disease control are essential tasks as they directly influence the performance of crop development. To realize effective weed control and to attain high yields, agrochemicals such as pesticides, herbicides, and fertilizers, are currently used in conventional agriculture. Figure 1.1 (left) depicts an everyday situation in a crop field. A tractor treats the entire field by uniformly spraying the same dose of agrochemicals to the soil,

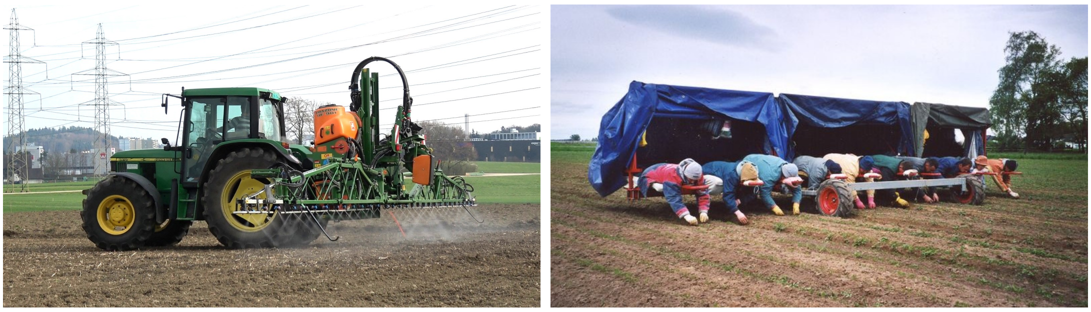


Figure 1.1: State-of-the-art in weed control. Left: uniform application of agrochemicals on a crop field. Right: low-throughput and expensive hand-weeding on organic farms [39].

crop plants, and weeds. This commonly applied practice is easy to execute for the farmer. It neither requires knowledge about the spatial distribution of plants and weeds nor about the type of weeds. Agrochemicals, however, can harm the environment, biodiversity, and consequently can affect human health [54]. We must, therefore, develop new methods that avoid today's agrochemicals or at least reduce their use to the necessary minimum.

Weed control in organic farms, such as shown in Figure 1.1 (right), is a labor-intensive and thus expensive task. Hand-weeding is still the current practice when it comes to the removal of individual weeds on organic farms. Among the chemical-free methods, this technique is the most effective way of preventing the weed from spreading. Tractor-based mechanical weeding tools do not yet provide the desired accuracy. Thus, in practice, weeding by hand is often additionally performed after tractor-based mechanical weeding to deal with weeds that are left in the crop row.

We believe that productive and future-oriented agriculture needs to focus on both high productivity and sustainability at the same time. One goal of sustainable farming is to reduce the reliance on agrochemicals while keeping the yield high. Precision farming techniques seek to address this goal. A promising but still to be explored way consists of first monitoring the field status by measuring key indicators of crop health as well as the spatial distributions of crop plants and weeds and second, providing targeted approaches for selectively treating only those plants that need it at the right time they need it.

1.1 Agricultural Robots to Advance Sustainable Farming

In crop production, a more effective and sustainable solution is to replace the current uniform spraying approach by more selective and targeted approaches. Autonomous agricultural field robots, such as the unmanned ground vehicle (UGV) illustrated in Figure 1.2, can perform continuous per-plant monitoring as well as selective and targeted treatments. For selective in-field treatments, robots will be equipped with different actuators for intervention, such as selective sprayers, mechanical tools, or even lasers, which perform plant-specific or species-specific treatments only at those locations where it is actually needed. Equipped with a variety of tools, robots can choose the most effective treatment based on the type of targeted plants and weeds. Consequently, UGVs offer an attractive solution for a drastic reduction of agrochemicals that are applied in the fields while limiting the operational costs [7, 8, 151]. To build autonomous agricultural robots for selective and plant-specific treatments, several open problems need to be solved.

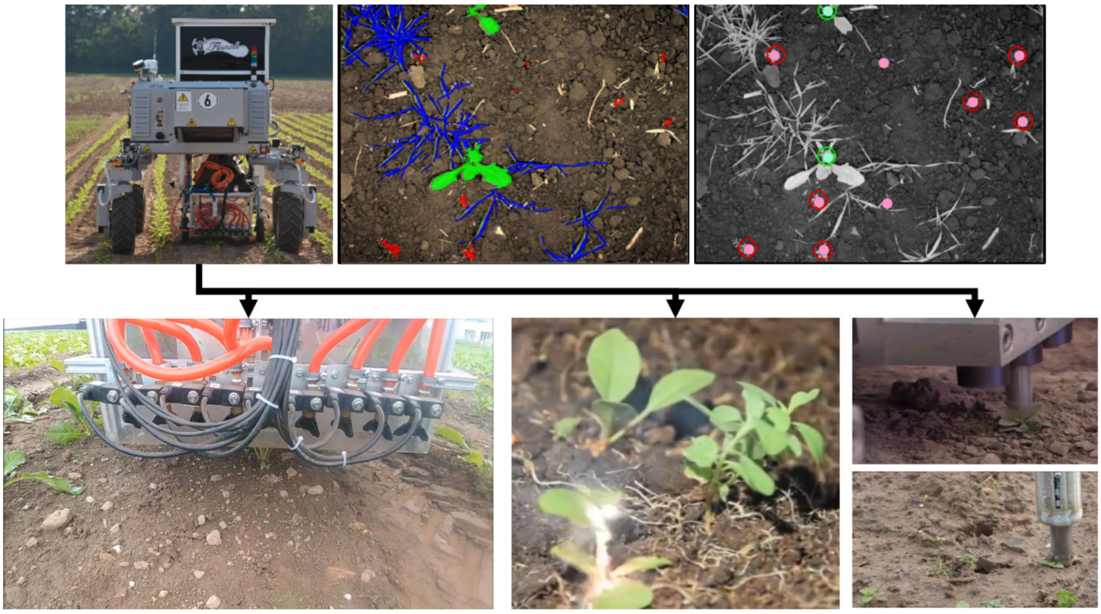


Figure 1.2: The BoniRob V3 field robot performs selective and species-specific treatment in the field. In this thesis, we present plant classification systems that analyze RGB and near-infrared images to provide the spatial distribution of plants and weeds. In this example, we consider the classes crop (green), dicotyl weed (red), grass weeds (blue), and soil. We provide the precise location of the plant and weed stems. With this information about weed coverage and stem positions, the robot can select the most appropriate treatment, such as spraying for grass weed, mechanically stamping or burning dicotyl weeds, or fertilizing the crop plants.

This includes robust perception, fast and effective actuators, rough terrain navigation, long-term autonomy, and several other factors [149].

This thesis tackles the central problem of robust perception in crop fields for autonomous farming robots. We investigate the challenge of how to interpret the image data recorded with a robot in the field. Figure 1.2 depicts one of the used UGVs during operation in the field. We propose several novel vision-based classification systems that provide the robot with information about the spatial distribution of the plants and weeds. Through this, we enable robots to trigger actuators at the right location at the right time to solve the desired task.

Another popular way to monitor farmland on a larger scale is through unmanned aerial robots (UAVs). UAVs can cover large areas in a comparatively short amount of time without interacting with the environment as ground robots do [25, 150]. Depending on the altitude and the used camera system, UAVs offer the freedom to capture image data on a rather coarse scale of a few centimeters ground resolution or to capture comparably dense information about field status in the range of sub-millimeter resolution. Thus, they are conceivable for several applications, e.g., on a larger scale for monitoring problem areas or the spatial distribution of plants and weeds in the field, but also for determining plant traits, monitoring specific weeds, or counting plants. This information supports

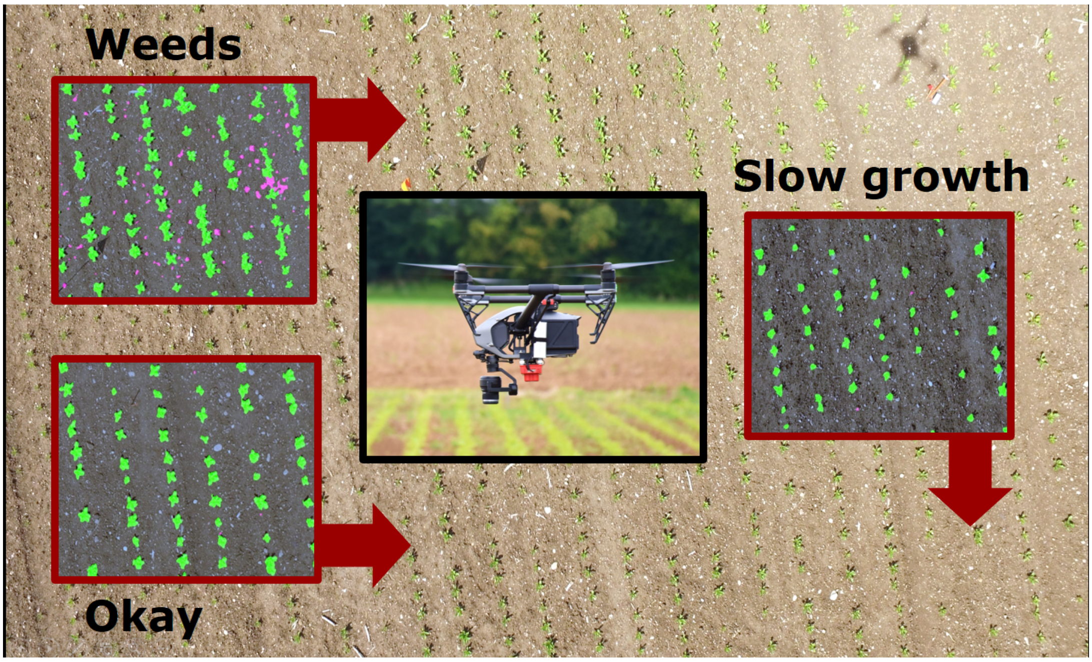


Figure 1.3: A quad-rotor UAV (DJI Inspire II) that we use in this thesis. We use the UAV to carry an RGB camera over the field to monitor the farmland. Then we analyze the image data with our plant classification systems to determine the spatial distribution of crop plants and weeds as well to identify traits such as the canopy cover and stand count.

the farmer in making decisions regarding field management and we can also derive application maps from this information to guide field robots in terms of weed control and fertilization.

In this thesis, we also investigate novel vision-based plant classification systems for processing UAV images. We propose systems which can classify crop plants and weeds as well as different weed species in UAV data. Furthermore, we propose an approach that can robustly count the number of plants, even under harsh field conditions, to provide relevant information to farmers or breeders in an automated manner.

1.2 Plant Classification Systems for Agricultural Robots

A prerequisite for any selective and plant-specific treatment is an effective plant classification system providing the robot with the locations of the plants and weeds in the field. The robot can use this information to trigger its actuators to perform the desired action in real time. For UAV-based crop monitoring, we relax the real-time online processing constraints for the developments in this thesis as we process the data offline, i.e., after the UAV has landed. Thus, for UAV data, we do not target on-board processing capabilities as it is the case for the UGV.

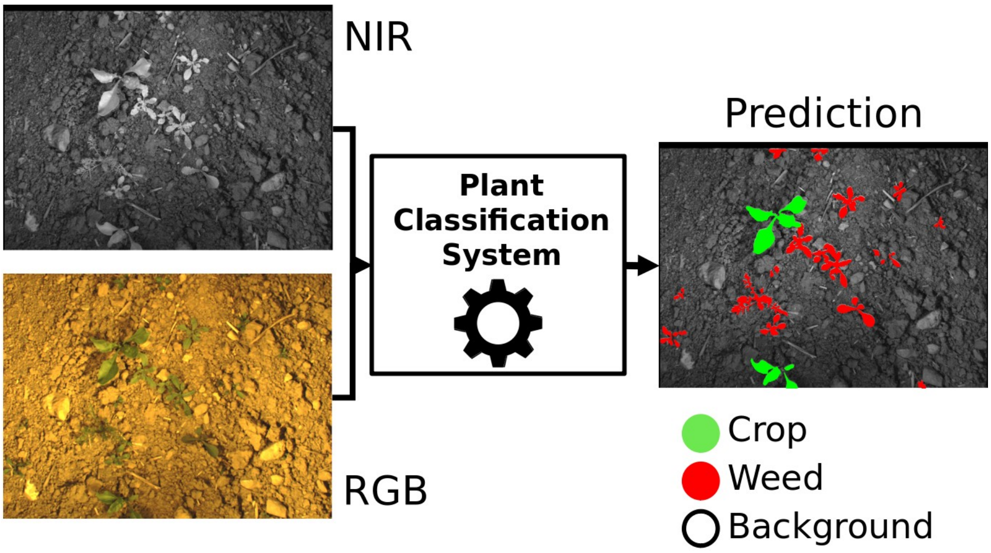


Figure 1.4: Goal of the plant classification systems in this thesis is to perform a pixel-wise classification. In the illustrated example, we consider the classes soil, sugar beet, and weed.

We use cameras to observe the environment and analyze the images regarding the presence and location of plants and weeds. We use RGB images and additionally utilize near-infrared (NIR) intensity measurements. The NIR information is especially useful for separating the vegetation from the soil and other backgrounds due to the high reflectivity of chlorophyll and thus plants in the NIR spectrum.

For an effective in-field intervention, it is crucial to be able to classify the crop plants and weeds throughout the entire growing season. For weed control, the earlier mechanical or chemical weeding actions are executed, the higher the chances for obtaining a high yield. In contrast, fertilization is carried out as long as vehicles can get into the field. Thus, classifications systems have to deal with a large variety of different growth stages of plants and weeds, but also with different soil conditions. These conditions mean that we are looking for objects that have a diverse appearance over time and can rapidly change their appearance due to environmental influences. We utilize machine-learning techniques that can learn automatically from experience and improve with new data examples, but without being explicitly programmed. We mainly investigate and use two types of machine-learning models: first random forests and second fully convolutional neural networks.

Figure 1.4 illustrates the principal goal of the vision-based plant classification systems in this thesis. The system receives image data for analysis. The task of the classifier is now to *assign a class to each pixel* in the output. In this thesis, we call this process pixel-wise classification. Note that in the literature, this form of classification is also called semantic segmentation.

Depending on the application, a plant classification system has to provide its output at different levels of complexity. With the term complexity, we refer to number of classes the model has to predict and the type of output for a given task. In the following, we give a few examples according to the addressed tasks in this thesis.

For the application of weed control in the pre-emergence phase of the plants, for instance, the classifier needs to provide the spatial distribution of the vegetation in the field, leading to a binary classification problem (vegetation vs. soil). Right after the emergence of plants, the complexity of the classification problem increases. The classifier now additionally needs to distinguish the crop plants from the weeds. This information is required so that the robot can selectively treat the weeds while it protects the crops from being eliminated. At the next level of complexity, the classification system needs to detect various sets of classes for even more sophisticated in-field operations such as species-specific treatments. Another layer of complexity is required for high precision interventions, such as precise mechanical and laser-based weeding, as these approaches are most effective when applied to the stem locations of small weeds. In contrast, big weeds and generally grass-like weeds are most effectively treated by spraying agrochemicals over their entire leaf area. For these high precision interventions, the classifier needs to provide the exact stem location within the level of a few millimeters to guide the robot's actuation system.

1.2.1 Challenges and Requirements for Vision-Based Plant Classification Systems

There exist several challenges for the development, deployment, and commercialization of plant classification systems for autonomous agricultural robots. Not only must the performance of the plant classification system be adequately high, but the classifiers must also be robust enough to work appropriately under changing field conditions. Plants can continuously change their appearance in size, color, and shape. Furthermore, the appearance of the field is affected by external circumstances such as weather events or cultivation processes. This means that the classification models must cover a high degree of heterogeneity in the data, but should also be efficiently adaptable to local field variations and unseen situations. In the remainder of this section, we present key practical challenges for vision-based plant classification systems and define properties that such a system should have.

Performance: An essential property of a plant classifier is a proper plant classification performance during its deployment. The obtained accuracy needs to be adequate for the given task. For example, in a weed-control scenario, it

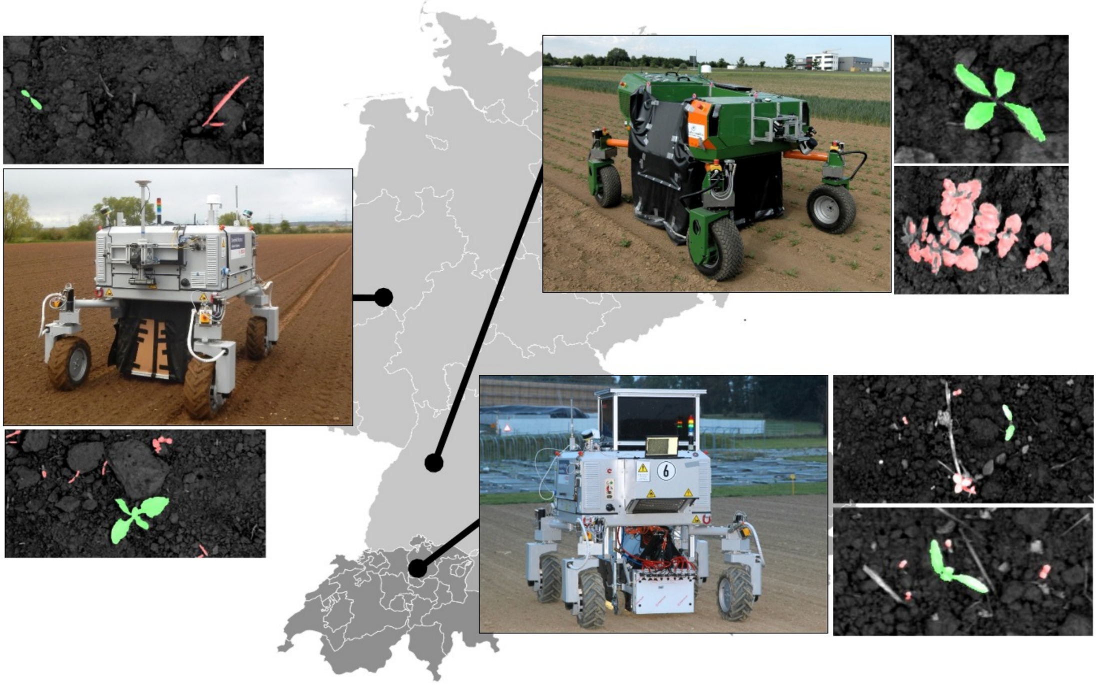


Figure 1.5: In this thesis, we aim at keeping the classification performance of the crop-weed detection system high, even if the training and the operational phase of the classifier are executed on different fields and in different countries as depicted in the image above. The example images are analyzed by one of our proposed plant classification approaches. Sugar beets (green) and weeds (red). We can even detect tiny plant and weeds from a size of 0.15 cm^2.

is crucial to know how many weeds are correctly identified as such. Another critical variable is how many plants are falsely identified as weeds and potentially eliminated by the robot. In the case of crop monitoring, the focus of the evaluation is more on the performance of the plants, e.g., how accurately can we count the plants or how precisely can we determine the size of the plants?

Generalization capabilities: In the last decade, several vision-based methods have been proposed for plant classification. Typically, such approaches are based on supervised machine-learning techniques and report classification performances in the order of 75-95 % in terms of classification accuracy. However, we see a lack in the evaluation of several methods regarding the generalization capabilities to unseen situations, new fields, and changing field conditions. Precision farming robots need to operate in different field environments regularly. A typical use case is that a plant classifier has been trained on data coming from one or more particular field environments, but is then deployed at a later point in time or in another field, where the visual appearance of the plants, weeds, and soil has notably changed. These changes can lead to different distributions of the image data and features concerning the original training data. In most cases, vision-based classification systems suffer under these conditions and provide insufficient performance. Slaughter *et al.* [137] conclude in their robotic weed control systems review that the missing generalization capabilities of the crop-weed classification technology constitute a major problem for the deployment and commercialization of such systems. The generalization capabilities to new field environments are essential for the actual deployment of farming robots for selective intervention and crop monitoring in the real world. Therefore, one focus of this thesis is the evaluation of the plant classification system under changing environmental conditions.

Labeling effort: A further challenge for supervised machine-learning approaches is the necessary amount of labeled data. Labeling training data for such approaches is a laborious task and thus expensive. To train a classifier that works with high performance in different scenarios, we rely on a sufficient amount of training data. Thus, approaches to reduce the amount of labeled data are of high relevance.

1.3 Goals and Main Contributions

The main objective of this thesis is the development of innovative vision-based plant classification systems for agricultural robots allowing the robots to identify the value crop and distinguish it from weeds or even different weed species. Our key developments focus on plant classification systems that enable UGVs for online, in-field interventions and enable UAVs to be used for accurate plant monitoring applications. We aim at improving the generalization capabilities to new and changing field conditions. To achieve robust performance in new field situations, we propose novel approaches that exploit that a large number of crop plants are sown in rows. Sugar beet plants, for example, are arranged in crop rows and often share a similar lattice distance along the crop rows. Such geometric information is typically similar within and across fields and, thus, less dependent on the visual appearance of the plants. To the best of our knowledge, this is the first work explicitly addressing the generalization capabilities of plant classifiers to new and changing field conditions through the development of novel approaches that exploit the spatial arrangement of the plants.

Under consideration of real-world applicability, we implement our methods for use on agricultural robots. We evaluate the developed approaches to real-world datasets in a thorough experimental evaluation. Therefore, we acquired an extensive and diverse database. The crops considered in this work are mainly sugar beets, an important row crop in Germany, and other countries of Northern Europe.

In the following, we describe the main contributions of this work along with an overview of our proposed approaches in Table 1.1. The table provides for each approach its name, an abbreviation, the application, and a small description.

Table 1.1: Overview of our proposed random forest-based and fully convolutional neural network plant classification systems. Each approach has been either entirely or partially presented in our published conference papers [82, 85, 86, 88, 89] or journal articles [83, 84, 87].

Description	Random Forest	FCN	Description
Visual Plant Classification Described in sections 4.3 and 5.2			
Keypoint-based approach classifying lattice-spaced keypoints	RF-KP [81, 86]	FCN [82]	Fully convolutional neural network for plant classification on single images
Object-based approach classifying connected vegetation components	RF-OBJ [87]		
Cascaded approach combining RF-KP and RF-OBJ	RF-CAS [87]		
		FCN-STEM [82]	FCN for plant classification and stem detection on single images
Visual and Geometrical Plant Classification Described in sections 4.4 and 5.4			
Geometric classifier exploiting plant arrangement	GC [89]	FCN-SEQ [84]	Sequential FCN for plant classification on image sequences
Semi-supervised approach exploiting visual RF-CAS and and geometric GC classifier	RF-GC [89]		
		FCN-SEQ-STEM [83]	Sequential FCN for plant classification and stem detection on image sequences
UAV-Based Plant Classification Described in sections 4.5 and 5.2.2			
RF-CAS exploiting geometric features for UAV imagery	RF-UAV [88]	FCN-UAV [85]	FCN for plant classification and stem detection on UAV images exploiting larger spatial context
		FCN-UAV-STEM	FCN-STEM applied crop counting and plant classification based on UAV imagery

1. A vision-based cascaded plant classification system based on handcrafted features using random forests. We call this approach RF-CAS. The random forest-based classifier combines two subordinated approaches that exploit two different ways to address the feature extraction for the classification problem. The first approach extracts local features for keypoints and classifies the area around each keypoint. We refer to this approach with RF-KP. The second variant is an object- or segment-based classification that determines all pixels in a vegetation segment. We refer to this approach with RF-OBJ. Please note that we do not claim a contribution to the RF-KP approach in this thesis as we developed this approach with the context of the master's thesis by Lottes [81]. However, we propose other approaches that build upon RF-KP. Thus, we also explain the RF-KP approach. Both approaches, RF-KP and RF-OBJ, have their advantages and disadvantages. RF-CAS combines RF-OBJ and RF-KP in a cascade and exploits their respective advantage and even compensates for their respective disadvantages.

2. A vision-based plant classification system based on a lightweight, fully convolutional neural networks. We call this approach FCN.

Both vision-based classification systems RF-CAS and FCN identify plants using RGB+NIR or RGB-only imagery as their input, can deal with small as well as overlapping plants, can solve a multi-class problem, are deployable on real agricultural robots, and provide the results of the classification fast enough for online in-field interventions with UGVs.

3. Adoption of the random forest-based RF-CAS approach and the fully convolutional neural network-based FCN approach for analyzing UAV data. We adapt the random forest-based by adding additional handcrafted features exploiting the crop row structure and spatial relationships between plants and weeds in the field. We adapt the fully convolutional neural network-based approach by modifying its network architecture. The modifications aim at enlarging the considered neighborhood in image-space that contributes to a prediction for a single pixel, thereby allowing the fully convolutional neural network to learn features describing the field geometry. We call these approaches RF-UAV and FCN-UAV, respectively.

4. A novel extension of the fully convolutional neural network-based plant classification system for pixel-wise plant segmentation to jointly detect plant stems enabling for high precision plant- and species-specific treatments such as shown in Figure 1.2. The system jointly estimates the pixel-wise segmentation into the classes crop, dicotyl weeds, grass weeds, and soil, and additionally provides the stem locations for the crop plants and dicotyl weeds

at the same time. We call this approach FCN-STEM and for UAV-based applications FCN-UAV-STEM.

5. A semi-supervised online learning approach for the random forest-based classification of crop plants and weeds by exploiting additional arrangement information of the plants in order to adapt the visual classifier to new and changing field conditions. We use a probabilistic model representing the arrangement of the plants and employ a Bayesian approach to perform the crop and weed classification only based on the geometric model. Then, we combine the visual classifier with the additional geometric classifier that complement each other within a semi-supervised online-learning scheme. Therefore, we modify the visual classifier to be suitable to perform online learning by using the predictions of the geometric classifier in order to adapt its model to the current situation in the field. We call this approach RF-GC and call the standalone geometric classifier GC.

6. A novel architectural extension to fully convolutional neural networks that classify plants based on analyzing image sequences. We refer to our proposed extension with the sequential module. It enables the usage of image sequences to encode features describing the local arrangement of the plants. Our approach exploits this geometric signal to improve the generalization capabilities of the plant classifiers. This technique leads to a better classification performance as well as to better generalization capabilities of the classifier, even if the visual appearance or the growth stage of the plants change between training and test time. We show that our sequential approach outperforms classification models which operate on single images. We call this approach FCN-SEQ for the extension of FCN and we call this approach FCN-SEQ-STEM for the extension of FCN-STEM.

7. For the evaluation of the proposed approaches, we collected a comprehensive database. It consists of approximately 26,500 labeled images captured by UGVs and UAVs, providing a full pixel-wise annotation of the class labels crop, weed, and soil. For almost 5,000 of those images, we additionally provide additional labels of grass weeds and the location of the plant and dicotyl weed stems. During the years 2015 and 2019, we acquired the data in different fields located in Germany, Switzerland, and Italy. We evaluate our approaches in the context of their performance, generalization capabilities, labeling effort, use of additional NIR information, and architectural design choices. For this thesis, we redid all the experiments that we previously published in papers on the identical database to ensure a fair comparison of the proposed approaches. Please note that the repetition of the experiments on the complete database may lead to deviations in the performance metrics

from the respective published published conference papers [82, 85, 86, 88, 89] or journal articles [83, 84, 87]. Besides the database, small deviations in the performance may occur due to changed parameter settings in the evaluation reported here.

1.4 Structure of this Thesis

This thesis is structured as follows: In Chapter 2, we present an overview of the used machine-learning models within this thesis, i.e., random forests and fully convolutional neural networks. Furthermore, we provide information about the used metrics for the evaluation of our proposed plant classification systems. In Chapter 3, we introduce the ground-based and aerial robots used for data acquisition and deployment in the fields as well as our comprehensive database used for development and evaluation of the classification systems. In Chapter 4 and Chapter 5, we present the different plant classification models. In Chapter 4, we describe the approaches based on handcrafted features using the random forest as their classification model. In Chapter 5, we describe the approaches based on fully convolutional neural networks. In both chapters, we describe the classification models, how we integrate the use of geometric features describing the plant arrangement into the classifiers, and how we adapt the classifiers to also work with UAV data. Chapter 6 includes our comprehensive experimental evaluation of the approaches presented in the approach chapters. In Chapter 7, we present related approaches in this field and set our work into the context of the related work. In Chapter 8, we present our conclusion and an outlook to potential future works.

1.5 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles, for which this thesis claims the main contribution:

- P. Lottes, M. Höferlin, S. Sander, M. Müter, P. Schulze-Lammers, and C. Stachniss. An Effective Classification System for Separating Sugar Beets and Weeds for Precision Farming Applications. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016

- P. Lottes, M. Höferlin, S. Sander, and C. Stachniss. Effective Vision-based Classification for Separating Sugar Beets and Weeds for Precision Farming. *Journal of Field Robotics (JFR)*, 34:1160–1178, 2017

- P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss. UAV-Based Crop and Weed Classification for Smart Farming. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017

- P. Lottes and C. Stachniss. Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017

- P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018

- P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 3:3097–3104, 2018

- P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Robust joint crop-weed classification and stem detection using image sequences. *Journal of Field Robotics (JFR)*, 37(1):20–34, 2020

as well as the following workshop paper:

- P. Lottes, N. Chebrolu, F. Liebisch, and C. Stachniss. UAV-based field monitoring for precision farming. In *25. Workshop Computer-Bildanalyse in der Landwirtschaft*, 2019

1.5.1 Collaborations

Parts of this work were part of different collaborations, which we acknowledge in the individual chapters, and have led to the following peer-reviewed conference and journal articles with:

- N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *Intl. Journal of Robotics Research (IJRR)*, 36(10):1045–1052, 2017

- A. Milioto, P. Lottes, and C. Stachniss. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W3, pages 41–48, 2017

- A. Milioto, P. Lottes, and C. Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018

- I. Sa, M. Popovic, R. Khanna, Z. Chen, P. Lottes, F. Liebisch, J. Nieto, C. Stachniss, and R. Siegwart. WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming. *Remote Sensing*, 10, 2018

- A. Walter, R. Khanna, P. Lottes, C. Stachniss, R. Siegwart, J. Nieto, and F. Liebisch. Flourish - a robotic approach for automation in crop management. In *Proc. of the Intl. Conf. on Precision Agriculture*, 2018

- N. Chebrolu, P. Lottes, T. Laebe, and C. Stachniss. Robot Localization Based on Aerial Images for Precision Agriculture Tasks in Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019

- A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebisch, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popovic, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer an R. Siegwart, C. Stachniss, A. Walter, V. Winterhalter, X. Wu, and J. Nieto. Building an Aerial-Ground Robotics Systemfor Precision Farming. *IEEE Robotics & Automation Magazine*, 2020

- X. Wu, S. Aravecchia, P. Lottes, C. Stachniss, and C. Pradalier. Robotic weed control using automated weed and crop classification. *Journal of Field Robotics (JFR)*, 37:322–340, 2020

- R. Sheikh, A. Milioto, P. Lottes, C. Stachniss, M. Bennewitz, and T. Schultz. Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020

as well as the following workshop papers:

- F. Liebisch, J. Pfeifer, R. Khanna, P. Lottes, C. Stachniss, T. Falck, S. Sander, R. Siegwart, A. Walter, and E. Galceran. Flourish – A robotic approach for automation in crop management. In *In Proc. of the Workshop für Computer-Bildanalyse und unbemannte autonom fliegende Systeme in der Landwirtschaft*, 2016

- F. Liebisch, M. Popovic, J. Pfeifer, R. Khanna, P. Lottes, C. Stachniss, A. Pretto, S. In Kyu, J. Nieto, R. Siegwart, and A. Walter. Automatic uav-based field inspection campaigns for weeding in row crops. In *Proceedings of the 10th EARSeL SIG Imaging Spectroscopy Workshop*, 2017

Chapter 2

Basic Techniques

This chapter describes the basic techniques and the framework for performance evaluation of our proposed plant classification systems. For the used machine-learning model, we recapitulate random forests, which were introduced by Breiman [15] in the year 2001 and fully convolutional neural networks. Furthermore, we give an overview of the used metrics to assess the achieved plant classification performance. Here, we also present two ways for the evaluation, i.e., a pixel-wise and an object-wise evaluation.

2.1 Machine-Learning Models

We define the input data for a classification model as a set of data points $\mathcal{X} = \{x_n\}_{n=0}^{N}$ and the corresponding classification output as $\mathcal{Y} = \{y_n\}_{n=0}^{N}$. Here, N refers to the number of data points. A classifier can be seen as a function

$$\mathcal{Y} = f(\mathcal{X}, \Theta) \tag{2.1}$$

that maps the input $\mathcal{X}$ to the desired output $\mathcal{Y}$. Here, $\Theta = \{\theta_d\}_{d=0}^{D}$ refers to a set of internal parameters of the classifier and reflects the variables we want to learn to perform the desired classification. For instance, the input $\mathcal{X}$ to the classifier can be given directly by raw sensor measurements, such as images, or by higher level representations such as extracted features. Then the classifier f maps $\mathcal{X}$ to the classification result $\mathcal{Y}$. In case of pixel-wise classification of an image, $\mathcal{Y}$ is again an image that encodes a class label for each pixel location of the input.

The process of classification can be split into the training phase and the deployment phase. In the training phase, the goal is to learn values for Θ to perform the desired mapping to the correct class labels. In the deployment phase, we freeze Θ and apply the "learned" mapping from $f\colon \mathcal{X} \mapsto \mathcal{Y}$.

In this thesis, we mostly perform the training of the classification models using supervised learning. This implies that we know a desired output y_n, so-called label, for each data point in the training dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$. Here, $\mathcal{X}$ is the input data and $\mathcal{Y}$ the corresponding ground truth information. We are interested in the classification of class labels $\hat{\mathcal{Y}}$. Both, random forests as well as convolutional neural networks provide the classification output in a probabilistic manner, i.e., a discrete pseudo probability distribution of scores over the considered class labels $p(\hat{\mathcal{Y}} \mid \mathcal{X})$. Thus, we can formulate the classifier as

$$p(\hat{\mathcal{Y}} \mid \mathcal{X}) = f(\mathcal{X}, \Theta). \tag{2.2}$$

Below, we describe the random forest classifier and, consequently, decision trees as well as convolutional neural networks as our mainly used classification models within this thesis. Both models can be represented by Equation (2.2), whereas their corresponding input data differs due to its representation. The input data to the random forest classification model is given by a set of handcrafted features representing a higher-level representation of the original input. For convolutional neural networks, the input is directly given by $\mathcal{X}$, which can be the raw sensor data, such as images, and additional information such as wheel odometry. In case of the feature-based representation, each data point to be classified is represented by a feature vector $\boldsymbol{v} = [v_0, \ldots, v_m]$ with $m = 0, \ldots, M$. Consequently, for N data points, we can also represent the input data by a feature matrix $\boldsymbol{V}$ of size $N \times M$. Each of the N rows in $\boldsymbol{V}$ represents a data point, whereas each of the M columns represents a certain feature v_m. Thus, the distribution $p(v_m)$ of a specific feature is given by the m^{th} column of the feature matrix $\boldsymbol{V}_{*,m}$.

2.1.1 Decision Trees

As a decision tree is the core building block of a random forest, we first describe this classification model discussing its principle, the training-, and the deployment phase. Decision trees can be seen as a hierarchy of consecutive applied decisions to the data alongside a tree structure.

Figure 2.1 depicts the flow of the data for two exemplary decision trees. The first node in the tree is the so-called root node. It receives all data points. Then the data passes decision nodes (sometimes also called split nodes) subsequently. Within a decision node, a certain decision rule $\theta_d(m)$ is computed to binary split the data into the corresponding child nodes. Thus, the root node itself is also a decision node. Finally, the data points are split into a leaf node. The latter are not further split and hold the classification results as each leaf node is mapped to a certain class label or a pseudo probability distribution over the class labels. A dataset can have many features and decision rules, and therefore the tree becomes larger and gets more complex. Independent of the size of a tree, however, the importance of the used features and relationships of the decision rules can be directly accessed and lead to an interpretable model. A decision tree ϕ maps the

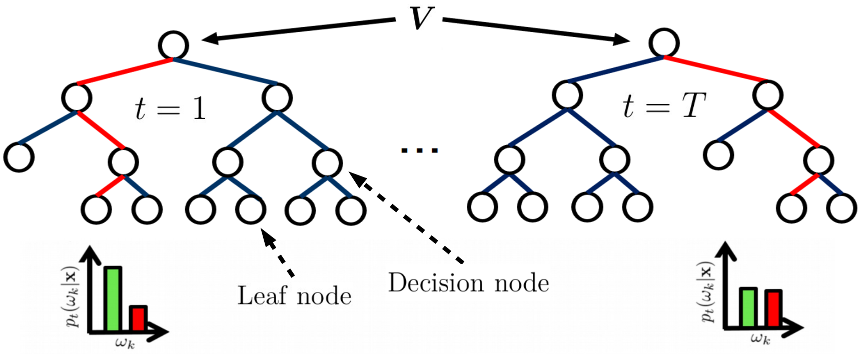


Figure 2.1: Random forests [15] are an ensemble learning method building a "forest" of T decision tree classifiers. Each decision tree is trained on randomly picked subset of the training data and provides a pseudo probability distribution over the possible class labels $\hat{\mathcal{Y}}$. According to Equation (2.6), the pseudo probability distributions of each decision tree are combined and represent the output of the random forest model, which is again a pseudo probability.

features for a data point to a pseudo probability distribution over the class labels, i.e.,

$$p(\hat{\boldsymbol{y}} \mid \boldsymbol{v}) = \phi(\boldsymbol{v}, \Theta). \tag{2.3}$$

Here, Θ refers to the set decision rules responsible for splitting the data and, thus, for the flow of the data through the tree-structured graph.

Building Decision Trees

Within the training phase, the goal is to learn Θ such that we obtain an efficient tree structure for the classification of new, previously unseen, data points. Here, efficient means that the learned tree provides the desired classification performance by a minimum number D of decision rules θ_d. We achieve this by a training procedure called recursive binary splitting. This procedure tries to greedily find the best possible data splits within each decision node, which maximizes the purity regarding the considered C classes $\hat{\mathcal{Y}}_c$ in the resulting child nodes. This step is repeated until only leaf nodes are left over. A split is performed by a split function

$$s(\theta_m) = \begin{cases} \text{left child}, & \text{if } v_m \leq \theta_d(m) \\ \text{right child}, & \text{otherwise} \end{cases} \tag{2.4}$$

as a threshold operation in feature space on a per-feature basis. To find the best split, we iterate over the respective feature distributions $\boldsymbol{V}_{*,m}$ given by the features along each column in the feature matrix and search for the $\theta_d(m)$ maximizing the purity of class labels in the child nodes. Instead of maximizing the purity, we use the gini-score G as measure for impurity in the child nodes and try to find the split with the minimum gini-score such as

$$G_{\theta_d} = \underset{m \in \{1,\dots,M\}}{\operatorname{argmin}} \; 1 - \sum_{1}^{C} p(\hat{\boldsymbol{y}}|s(\theta_d(m)))^2 \tag{2.5}$$

as a measure of the impurity in the child nodes given the split. Here, $G(\theta_d)$ measures how "mixed" the class labels are in the child nodes.

An important question when building decision trees is: when to stop the training procedure? In other words, how to define a node as a leaf node? The most naive approach is to split the data in the nodes until 100 % of the data points correspond to a single class. This, however, leads to very deep tree structures and causes the model to over-fit to the training data as also noisy data points and outliers are split until all nodes reach a purity of 100 %. We can address this issue by the following techniques aiming at the early stopping of the training or pruning the tree post-training:

- **Minimum purity :** Defining a node as a leaf node by postulating a particular purity for the class labels by $p(\hat{\boldsymbol{y}}) \leq 1$.
- **Minimum samples:** A node is defined as a leaf node if the number of data points within the node is smaller than a defined threshold.
- **Maximum depth:** Stopping the splitting procedure when a branch in a tree reaches a certain depth, e.g., a defined number of splits to reach the leaf node.
- **Pruning:** After the training finishes, we prune each branch to either a certain depth or remove a specified number of the learned splits from the bottom up.

All these techniques have in common that they try to avoid overfitting the model to the training data. Thus, to reduce the generalization error by better performance on held-out test data and to the cost of a higher training error. The choice of the method(s) and corresponding parameters for controlling the size of the decision tree is a hyperparameter and needs to be evaluated empirically to find the optimum given the training and validation dataset.

2.1.2 Random Forests

Random forests are an ensemble learning method for both classification and regression. In this thesis, we focus on classification tasks and therefore describe this model only in regards to this purpose. The key idea of random forests is to construct a set of T decision-tree classifiers building the "forest" and to perform the classification as a majority vote based on the individual classifications made by the decision trees.

Building Random Forests

In the training phase, the goal is to learn T decision trees representing the random forest classifier. The term T is a hyperparameter and has to be set by the user. The basic principle of training the individual decision trees stays the same as described in Section 2.1.1 but with two modifications, called bagging and random feature selection, leading to effective characteristics of random forests.

Bagging (bootstrap aggregating): The first modification aims at randomizing the training data used to learn the individual decision trees. Given the training data $\mathcal{D}$ of size N, we randomly generate T training datasets $\mathcal{D}^s$ of size N^s for each decision tree by uniform sampling from $\mathcal{D}$ with replacement. This leads to a random forest that is given by

$$\Phi(\boldsymbol{V}, \Theta) = \{\phi_1(\boldsymbol{V}_1^s, \Theta_1^s), \ldots, \phi_t(\boldsymbol{V}_t^s, \Theta_t^s)\}_{t=1}^{T}. \tag{2.6}$$

By bagging, each decision tree is trained on a small but different portion of the training data, respectively. Thus, each decision tree learns a different set of model parameters Θ_t^s resulting in a set of weak learners, which perform slightly different predictions for the same test data points.

A further advantage of the bagging approach is that around 30 % of the training data points are never sampled by chance. Thus, these samples are not considered by any of the trees in the forest [15]. The random forest model stores the indices of those unused samples and uses them as validation data implicitly. These data points can then be used to provide metrics for early stopping techniques of the training procedure or as an intrinsic estimate for the generalization performance of the model.

Random feature selection: Given M different feature types, we randomly select a number of $M^s \ll M$ features to find the best data split given the minimum Gini-score in the corresponding child nodes according to Equation (2.5). Thus, it is not guaranteed to find the optimal split concerning maximizing the purity of the class labels within the child nodes, as smaller M^s leads to less similarity in the tree structure and less correlation between individual trees. Regarding Breiman [15], M^s is the only hyperparameter of a random forest.

Both methods bagging and random feature selection lead to more diversity in the individual decision trees in terms of the tree structure and learned decision rules. The random forest exploits the diversity of the individual decision trees as it performs the final prediction as a majority vote of its decision trees. Based on the outputs of the individual decision trees, we can compute a pseudo probability distribution for a single data point over the possible class labels $\hat{\boldsymbol{y}}$ by

$$p(\hat{\boldsymbol{y}} \mid \Phi(\boldsymbol{V}, \Theta)) = \frac{1}{T} \sum_{1}^{T} p_t(\hat{\mathcal{Y}} \mid \phi_t(\boldsymbol{V}_t^s, \Theta_t^s)), \tag{2.7}$$

where the elements in $\hat{\boldsymbol{y}}$ sum up to 1. Thus, random forests represent a multi-class classification model providing a probabilistic output reflecting the confidence of the classifier's predictions for data points belonging to the considered class labels. Through joining the respective predictions of the individual and diverse weak learners, random forests implicitly reduce the risk of over-fitting to some degree and are comparably robust to corrupted class labels within the training data. These properties make random forests attractive to use compared to other classical machine-learning models.

2.2 Neural Networks

Neural networks are the model that underlie deep learning. Many prominent deep learning models, such as convolutional neural networks and recurrent neural networks, can be represented by constrained or modified neural networks. Given a mapping $\mathcal{Y} = f^*(\mathcal{X}, \Theta)$, the main objective of neural networks is to approximate f^*. Neural networks typically consist of many simple processing units called neurons. They are connected within the network and communicate by receiving and responding signals via weighted connections. Neurons process the incoming data and realize the information flow through the entire neural networks. Inspired by biological neural networks in the human brain, neurons are sometimes also called perceptions, and consequently, neural networks are also called multi-layer perceptions.

Within a neural network, neurons are organized in layers. Figure 2.2 depicts a neural network consisting of $L = 3$ layers composed in a subsequent order. Note that in this example, the neurons are only interconnected between layers. This common structure is called a feed-forward network. The network in Figure 2.2 has a depth of three, as it consists of three consecutive layers, and has a width of three, as it has three neurons within each layer. The layer $f^{(l)}$ is called the l^{th} layer of the network. The first layer $f^{(1)}$ in a network is also called the input layer as it directly operates on the input $\boldsymbol{x}$, whereas the last layer $f^{(L)}$ is called the output layer. In-between layers are called hidden layers.

The j^{th} neuron in the network receives as input a set of responses created by I other neurons. These responses are so-called activations $\mathbf{a} = \Theta = \{a_i, \ldots, a_I\}_0^I$. Each a_i is associated with weight parameter w_i to control its influence for the

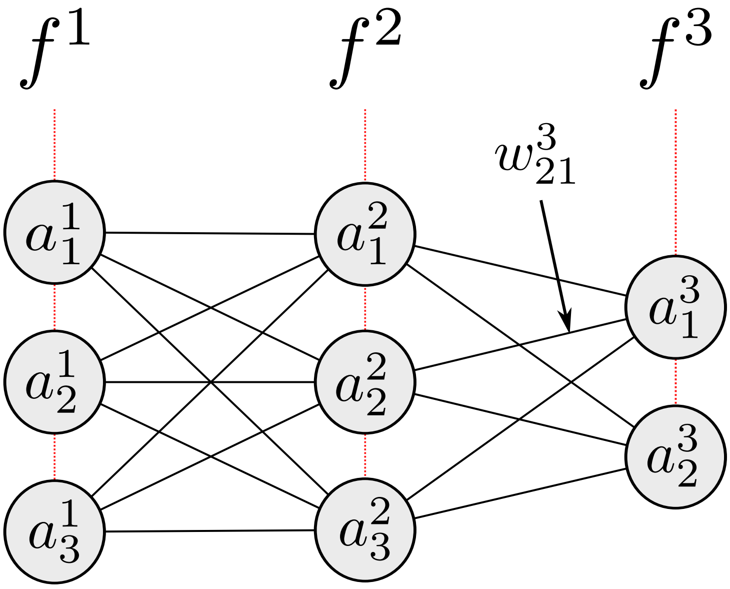


Figure 2.2: Illustration of three layer $L = 3$ neural network.

computation in the j^{th} neuron, i.e.,

$$a_j = \sigma\left(\sum_1^I w_i a_i + b\right) = \sigma(\mathbf{w}^T\mathbf{a} + b). \tag{2.8}$$

Here, b is an additional bias parameter and σ refers a non-linear activation function of the neuron. The latter we describe in Section 2.2.1. The values for w and b influence the information flow in the network and reflect the parameters that have to be learned to obtain the desire mapping of the input to the target variables. In this section, we substitute the learnable parameters of a neural network with Θ. The l^{th} layer in the network can be formulated as

$$\mathbf{a}^l = \sigma(\sum_1^J w_{ij}^l a_{ij}^{l-1} + b_j^l). \tag{2.9}$$

Written as a matrix-vector product, we obtain

$$\mathbf{a}^l = \sigma\left(\mathbf{W}^T\mathbf{a}^{l-1} + \mathbf{b}\right), \tag{2.10}$$

where $\mathbf{a}^l$ is the vector of resulting activations that can also be seen as the features computed by the l^{th} layer. The variable $\mathbf{W}$ refers to the weight matrix, and $\mathbf{b}$ refers to the bias vector. Thus, the entire graphical model in Figure 2.2 can be formulated as

$$\mathbf{a}^L = f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x}, \theta^1), \theta^2), \theta^3). \tag{2.11}$$

2.2.1 Activation Functions

The element-wise applied activation functions σ are standard building blocks of neural networks and allow us to define non-linear mathematical models. Only by integrating these functions the network can approximate a non-linear function and establish complex relationships between the input variables and the target variables. Concerning the optimization of the model parameters of neural networks, several non-linear activation functions can be chosen. However, in the past, several works have shown that it is best practice to use the rectified linear unit, i.e.,

$$\sigma_{ReLU}(a) = \max(0, a), \tag{2.12}$$

for the training of neural networks. Neural networks using the rectified linear unit activation function are less affected by the problem of the vanishing gradient and typically show better convergence behavior. In this thesis, we use the rectified linear unit as our standard activation function in neural networks.

The goal of a neural network-based classifier is to predict a discrete pseudo probability distribution over a considered number of class labels. Here, the number of neurons in the output layer corresponds to the number of classes. Thus, in our described toy example in Figure 2.2, the network considers two classes. We apply a softmax function to the last activations $\mathbf{a}^L$

$$\sigma_{\text{softmax}}(\mathbf{a}^L) = \frac{e^{\mathbf{a}^l}}{\sum_{j=1}^{C} e^{\mathbf{a}^l}} \ . \tag{2.13}$$

to obtain a pseudo probability for each class for a particular data point, i.e.,

$$p(\hat{\boldsymbol{y}} \mid \boldsymbol{x}, \theta) = \sigma_{\text{softmax}}(\mathbf{a}^L). \tag{2.14}$$

The softmax function a non-element-wise activation and transforms the output into a discrete pseudo probability distribution over the class labels The elements in $\hat{\boldsymbol{y}}$ sum up to 1.

2.2.2 Loss Function

The training of a neural network refers to the optimization of the model parameters concerning a loss function $\mathcal{L}$. In the case of supervised training, we have access to the input data $\boldsymbol{x}$ as well as to the actual labels $\boldsymbol{y}$. The loss function $\mathcal{L}$ computes the discrepancy between the predictions of the network and the labels. The goal of the training procedure is to adapt the model parameters Θ to minimize $\mathcal{L}$. As the loss function penalizes false classifications, the training should lead to an adaption of the parameters such that the network produces an appropriate mapping between the input and the output.

Several loss functions are considered to train neural networks. In this thesis, we use the weighted cross-entropy-loss for the training of our network-based classification models. It has the form:

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{c=1}^{C} \alpha_c y_c \log \hat{y}_c. \tag{2.15}$$

The cross-entropy evaluates the consistency of the ground truth and predicted distributions across the class labels. The variable α_c reflects an additional weight parameter and allows to penalize errors differently concerning their class label.

2.2.3 Gradient Descent

One of the key properties of neural networks is that they represent a fully differentiable function, as the entire network architecture is composed of small, but differentiable functions. Thus, we can perform a forward pass, i.e., feeding the input into the network and infer the output given the current configuration of parameters in the network structure. The resulting loss, according to Equation (2.15), reflects the signal we want to minimize. Therefore, gradient descent is used to identify the direction of the steepest descent in the parameter space. The algorithm computes the gradient of the cost function concerning Θ. The minimization of the loss function is therefore iterative given by

$$\theta' = \theta - \epsilon \nabla_\theta \mathcal{L}(\theta) \ . \tag{2.16}$$

The parameter ϵ denotes the learning rate telling the algorithm how far the step and, thus, how large the parameter updates are. The learning rate is one of the most important parameters to tune. A too small learning rate leads to too little learning progress, whereas a too-large learning rate adversely affects the convergence behavior or even lead to a divergence of the learning process.

In practice, the computation of the gradients over the entire training dataset is computationally expensive and thus impractical. Therefore, the parameter updates are performed batch-wise based on an iterative random sampling of the entire dataset. Such a sample is a so-called mini-batch, and the number of samples within a mini-batch is the so-called batch size B. After the network has processed all mini-batches representing the entire dataset, it has been trained over an epoch. Commonly, we train a model over several epochs, through a random sampling of the mini-batches in each epoch. The process is often called stochastic gradient descent.

2.2.3.1 Batch Normalization

Concerning the optimization of the model parameters, two problems often arise. One is the so-called vanishing gradient. Here the model parameters of deep network structures are not sufficiently optimized in the first layers due to too-low, vanished, gradient signals. The second problem is that during batch-wise training, the distribution of the activations can change dramatically and prevent the weigh parameters to converge to a stable solution. To address this problem, Ioffe and Szegedy [56] introduce batch normalization of the features in a batchwise manner during training. Batch norm tries to equalize the features by setting their mean to zero and scaling them by forcing their standard deviation to be one.

The normalization is followed by a learnable scaling and shifting of the feature distribution via two further layer-specific model parameters. Through the second scale and shift operation, the model can determine the optimal scaling and the mean value for the features that are inserted into the next layer. Additionally, batch normalization counteracts the model from overfitting [56, 52].

2.2.3.2 Dropout

Training a model on a limited amount of data comes at the risk of overfitting. To address this challenge, Srivastava *et al.* [138] propose dropout. Dropout randomly omits a particular amount of connections between neurons during the training training. Thus, not every neuron contributes to the processing of the current input.

This technique has different effects. First, dropout acts as a kind of generalizer, since different neurons in the network are forced to process the same information partially independently of each other, creating the same output. This leads to more general features, as the neuron is not allowed to rely on the presence of other neurons. Another view is that we train several networks at the same time, which then leads to a better during the deployment phase. Secondly, dropout means that information cannot always be processed in the same way by the network. This reduces the risk of the network to overfit to the training data by simply “memorizing” the correct solution.

2.3 Fully Convolutional Neural Networks

So far, we discussed neural networks with fully-connected layers, where every neuron of a particular layer is connected to all other neurons of the previous and subsequent layers. Classic feed-forward networks with fully connected layers use general matrix-vector multiplication, which can be computationally demanding, especially for the processing of high dimensional image input.

Convolutional neural networks have a similar structure to such networks in terms of layer-wise network architecture. However, they are mainly designed to process image data through the exploitation of convolutions instead of using fully-connected layers. Images have a particular topological structure that is given by the height H, the width W, and the channels or depth D of an image. Often, neighboring pixels correlate within an image. Convolutional neural networks explicitly exploit this information for the estimation task at hand.

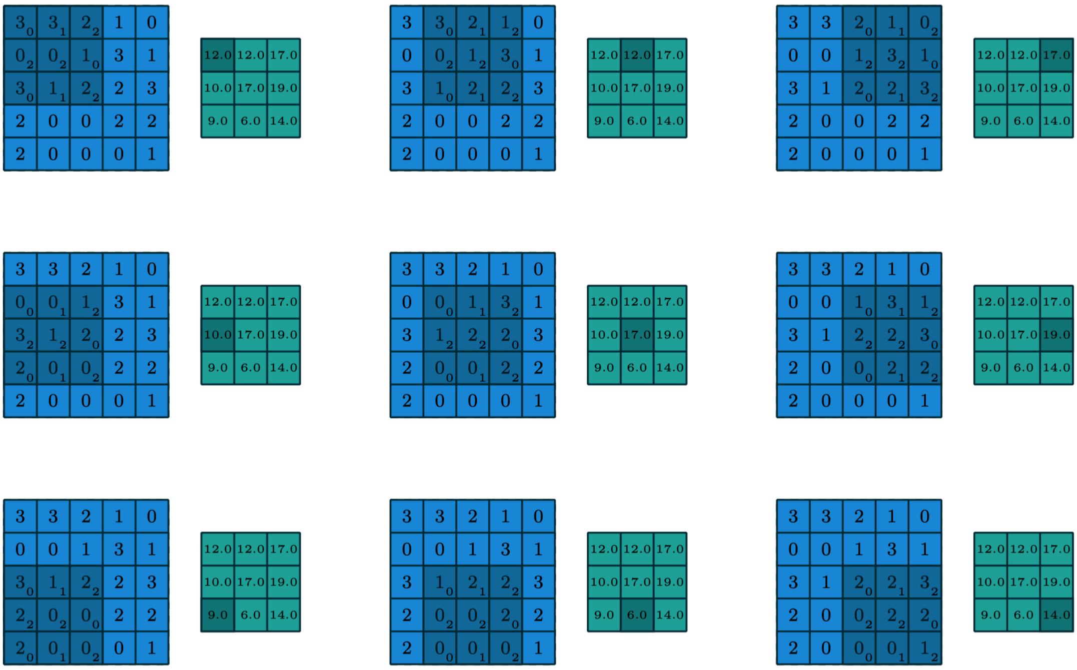


Figure 2.3: The dark blue convolutional kernel slides along the input in bright blue and sequentially performs a linear transformation of the input (green). Here, a 3×3 kernel containing the learnable weight parameters convolves the input of size 5×5. Through the valid convolution the output size is given by 3×3. Image courtesy of [27].

The term “fully convolutional” indicates that a convolutional neural network is an ensemble solely consisting of convolutional operations. Even for the last layer in the network, i.e., the decision-making layer, the operation is expressed through convolutions.

2.3.1 Convolutional Layers

The underlying mathematical operation in a convolutional layer is a discrete convolution. Figure 2.3 illustrates the process of the convolutional operation that is performed with the same filter over the whole input by sliding the kernel with a particular stride across the input. The result of that operation at one position of the input is the dot product of the kernel with the corresponding input region. Convolutions are commonly followed by an activation function, e.g., a rectified linear unit, and a batch norm operation that is performed channel-wise. This intermediate result is a so-called feature map.

In convolutional neural networks, several different kernels are applied to the same input. Each kernel produces a different feature map based on its respective weight parameters. All feature maps that are produced in one layer of the network form a feature volume. A feature volume represents the output of a certain layer. At the same time, it also represents the input to the consecutive layer

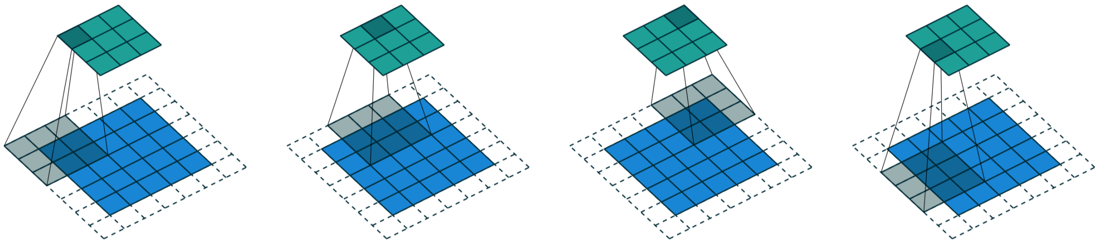


Figure 2.4: Zero-padding adds a row of zero elements to each boarder. The strided convolution with stride 2 then downsamples the input. Image courtesy by [27].

in the network. A convolutional neural network architecture consists of several subsequently applied convolutional layers that operate on the respective feature volumes produced by the previous layer. The number of subsequently applied layers defines the depth of the entire network.

Figure 2.3 shows that naively applying a stack of subsequent convolutions shrinks the output size of the respective features maps, as one is going deeper in the network. To control the spatial dimensions height H and width W of the features maps along with the depth of the network, we can use zero paddings of the feature maps to maintain their dimension. See Figure 2.4 for an example of zero paddings to keep the same H and W for a convolved feature map. This operation allows for the design of deeper architectures by stacking more layers. This, in turn, can lead to a better expressiveness of the model.

Typically, the spatial resolution of feature volumes is intentionally reduced with increasing network depth. This procedure enables the design of a deeper architecture concerning memory requirements. Furthermore, the successive reduction of the spatial dimensions leads to a higher information density of the computed features. Downsampling the feature map dimensions can be achieved through different methods such as max pooling or average pooling. Max pooling, for instance, is a kernel operator that copies maximum feature value into the downsampled feature map, whereas average pooling computes the average feature value. However, we use stridden convolutions in this thesis. The stride is the distance between two successive pixel positions of the filter in the input. The larger the stride, the lower the spatial resolution of the resulting feature volume. See Figure 2.4 for an example. In contrast to the pooling operators, the stridden convolution has trainable kernel parameters. Thus, it serves as a learnable downsampling of the features.

A particular convolutional layer in a convolutional neural network accepts an input feature volume of size $H_{in} \times W_{in} \times D_{in}$. It produces a feature volume of size

$$\begin{aligned} H_{out} &= \frac{H_{in} - K + 2P}{S} + 1, \\ W_{out} &= \frac{W_{in} - K + 2P}{S} + 1, \\ D_{out} &= F, \end{aligned} \tag{2.17}$$

with F being the number of convolutional filters and K their spatial extent, i.e., the kernel size, P denoting the amount of zero-padding and S the stride. Thus, the number of learnable parameters is given by $KKD_{in}F$. Note that we do not consider additional bias parameters. In this thesis we refrain from using additional bias parameters in our approaches, since these become superfluous due to the additional shifting parameters of the batch normalization operation, see Section 2.2.3.1.

The number of parameters is significantly reduced compared to neural networks using fully-connected layers, as the same filter is applied over the entire input. Therefore, the convolutional neural networks are less complex and have a lower risk of overfitting to the training data [60].

2.3.2 Encoder-Decoder Structured Networks

In a forward pass through the network, we first process the input by consecutive convolutional layers and downsampling to a smaller spatial resolution using stridden convolutions. This part of the network can be seen as the encoder and can be used as a feature extractor. The extracted features represent a spatially compressed but highly informative representation of the input. For the task of pixel-wise classification, however, we process the encoded features such that the output of the network is of the same spatial dimensions as the input. This part of the network can be seen as the decoder for the features and can be used to compute the desired pixel-wise classification output. In the decoder, we process the encoded features with convolutional layers and spatial upsampling operations to restore the original resolution of the input. Analog to the downsampling in the encoder, we perform the upsampling using stridden convolutions to realize the spatial enlargement of the features in a learnable way. Here, we use the so-called transposed convolution that represents the opposite operation of a convolution [27] compared to fully-connected neural networks. The output is finally given by a classification, which assigns a class to each pixel in the input.

2.4 Evaluation Metrics

In this section, we describe metrics that we use throughout the thesis for the evaluation of different properties of the investigated vision-based plant classifica-

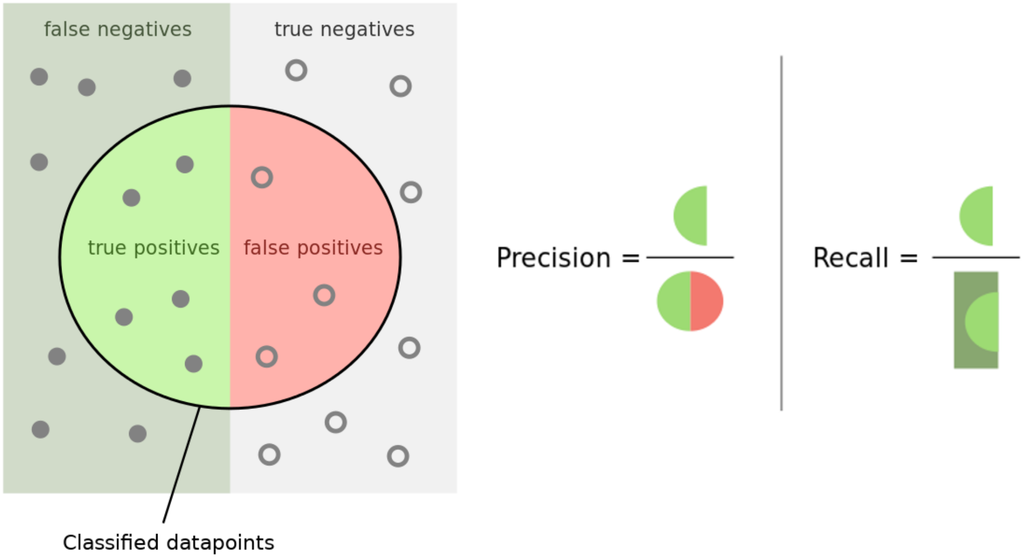


Figure 2.5: Precision and recall for a binary classification problem. Courtesy: *Wikipedia*.

tion systems. We provide further information about the experimental setup and evaluation strategies in our extensive experimental evaluation in Chapter 6.

Precision, Recall, and F1-Score

The precision-recall curve is a standard method to evaluate the performance of a classifier and is widely used in the field of computer vision, machine learning, pattern recognition, and information retrieval. The goal of the analysis of the precision and recall of a classifier is to asses its performance beyond the achieved classification accuracy, i.e., measuring how many data points the classifier identified correctly concerning all data points.

Consider a classification problem of crop against non crop. We want to classify each pixel in an image to determine whether it belongs to the class crop. Figure 2.5 illustrates how precision and recall a defined for this case. For the following explanations, we assume the crop to be the positive and non crop to be the negative class, i.e., green referring to crop and red referring to non crop.

The circles in Figure 2.5 refer to a single data point, e.g., an image pixel in the case of pixel-wise classification. All filled circles refer to actual crop pixels in the ground truth data, whereas all unfilled circles refer to actual non-crop pixels in the ground truth data. The correctly classified crop pixels are located in the green area of classified data points in Figure 2.5. As crop refers to the positive class in this example, these samples are called true positives (TP). The correctly classified non-crop pixels are called true negatives (TN) and located in the red area of classified data points. The classifier also produces missclassifications. Here, predicted crop pixels that actually belong to the non-crop class are called false positives (FP). Consequently, predicted non-crop pixels that actually belong to the crop class are called false negatives (FN).

The precision (P) for the crop can be seen as the probability that a randomly selected classified crop data point actually belongs to the crop class. Thus, the precision measures how likely it is that a classifier's prediction is correct. The term precision is defined as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.18}$$

The recall for the crop can be seen as the probability that a randomly selected actual crop data point from the ground truth is correctly predicted. Thus, the recall measures the number of actual crops that are found by the classifier. The term recall is defined as:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.19}$$

Both the precision and recall values have their minimum at 0 and their maximum at 1. Throughout this thesis, we provide the values for precision and recall in percent ranging from 0 % to 100 %. The precision and recall values have a direct meaning for the expected performance in the real world. For instance, a pixel-wise classifier that obtains a recall of 90 % for the crop class detects 90 % of all crop pixels in the data correctly. The precision gives an intuition on the ratio of hallucinated crop pixels. A classifier that obtains a rather low precision for crop predicts too many pixels as crop that are actually not crop.

For the comparison of two different classifiers, the accuracy of use and the recognition value is disadvantageous, because we have to consider two different values. Maybe one classifier has a high recall, but a low precession and the competitor has high precision, but low recall. So, we might want to ask: Which classifier is the better one? One way to approach an answer is to use the so-called F1-score, which is the harmonic mean of precision and recall, i.e.,

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.20}$$

The harmonic mean penalizes extreme values and gives an appropriate measure of imbalanced class distributions. Thus, the F1-score provides a single metric describing the overall performance of a classifier that takes both precisions and recall into account. Furthermore, the F1-score implicitly deals with imbalanced class distributions to some degree. Therefore, we use the F1-score for the comparison between classifiers throughout this thesis.

The above example deals with the calculation of metrics for a binary classification problem. However, in this thesis, we mainly deal with multi-class problems.

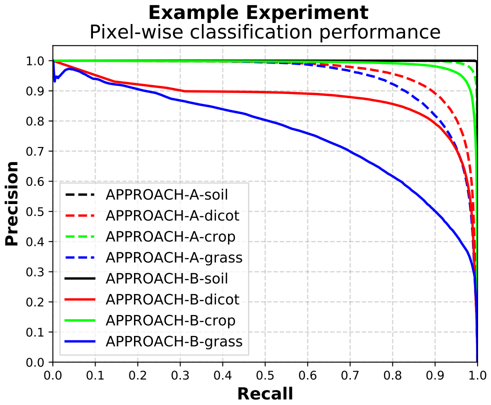


Figure 2.6: Example of a precision-recall plot comparing different approaches. We compare two approaches, called APPROACH-A and APPROACH-B, that predict classes crop, dicot-weed, grass weed, and soil.

In this case, we calculate the average of the class-wise measures, i.e., precisions recalls, and F1-scores across all considered classes, i.e.,

$$\operatorname{avg} M = \frac{1}{C} \sum_{\omega} M_{\omega}. \tag{2.21}$$

Here, M_ω acts as a placeholder for the class-wise precision, recall, or F1-score, and C the total numbers of considered classes. By averaging across all class metrics, we weight all classes equally, without taking into account their relative frequency of occurrence in the data.

A characteristic of random forests and convolutional neural networks is their probabilistic output. This means that we obtain a probability distribution $p(\omega \mid \text{model})$ over the considered class labels for every predicted data point. The evaluation of these probabilities offers a more detailed insight into performance than just comparing the final class label with the ground truth information. This also allows us to somewhat track the stability of the predictions. For multi-class problems, we first binarize the original multi-class task to multiple one-vs.-all classification problems. Through this, we always obtain one positive class ω_p and one negative class ω_n. Then, we create the class-wise precision-recall curves by varying a threshold $t \in [0, 1]$ for the mapping of the predicted probabilistic output $p(\omega \mid \text{Model})$ of the classifier to the assigned class labels, i.e.,

$$\omega = \begin{cases} \omega_p \text{ (positive)}, & \text{if } p(\omega_p \mid \text{Model}) \geq t \\ \omega_n \text{ (negative)}, & \text{otherwise} \end{cases} \tag{2.22}$$

By changing the parameter t from its default value 0.5, we can influence the classifier to either minimize false negatives or false positives. This allows us to visualize one precision-recall curve for every considered class.

Figure 2.6 depicts eight different precision-recall curves evaluating the performance of two different approaches for four different classes, respectively. The classifiers are called APPROACH-A and APPROACH-B and predict classes crop, dicot-weed, grass weed, and soil. Perfect classifier performance is given by a recall of 1 at a precision of 1. This means, the closer a precision-recall curve approaches the upper-right corner in the diagram, the better is the performance of the classifier.

First, we can visually get an impression of the class-wise performance of both classifiers. The class-wise precision-recall curves reveal that both classification approaches deliver an almost perfect performance for the classification of the soil class. Furthermore, both algorithms can provide crop plants with a recall of about 95 % at a precision of about 95 %. The performance for the two weed classes dicot-weed and grass weed is below this performance. In addition, we can compare the two classifiers at a glance. We see that the FCN-SEQ-STEM classifier achieves better performance for all plant species when compared. This can be seen from the fact that the respective class-wise curves are closer to the upper-right corner. The highest performance gain of FCN-SEQ-STEM is observed in grass weeds, where the distance between the two blue curves is most considerable.

In this thesis, we use precision-recall curves for most experiments to study the performance of individual approaches in detail and compare different classifiers. In an online application in the field, however, it is difficult to choose an optimal parameter t that is particularly suitable for the data. Therefore, during the deployment of the classifiers, we assign the class label concerning a labeling with the most likely class label that is predicted by the classifier, i.e.,

$$\omega^* = \underset{\omega}{\operatorname{argmax}}\, p(\omega \mid \text{model}). \tag{2.23}$$

For the class assignment with this method, we get one value for the precision, recall, and the F1-score. Besides the precision-recall curves, we always report the achieved performance that is achieved under a class labeling according to Equation (2.23).

Chapter 3

Agricultural Robots and Datasets

A prerequisite for any selective and targeted in-field treatment executed by an agricultural robot is a robustly working and high-performance plant classification system. Also, for crop monitoring applications with UAVs, a reliably working classification is an essential component within the process of automated data analysis. In this chapter, we present the UGV and UAV platforms used in this thesis. We describe the sensors for image acquisition of the UGVs actuation system for in-field weeding operations.

To cope with the large variety of different crops and weeds as well as with inherently changing environmental conditions, we need machine-learning techniques that can perform the mapping from the raw sensor input to the desired class labels. To exploit the full potential of machine-learning algorithms for the plant classification, a large and diverse database is essential for the development of high-performance plant classifiers, which are also robust and reliable under various field conditions. Existing and publicly available datasets for the plant classification, however, are often limited to a single or a small variety of aspects, e.g., single crop, a small amount of data, same sensor setup. They are typically collected at low frequency, or even at a unique point in time. In this chapter, we present an overview of the datasets that we collected within the Photogrammetry and Robotics lab in Bonn for the development and evaluation of this thesis.

In the years 2015 to 2019, we collected an extensive database along with this thesis including data (i) collected with different UGV and UAV platforms, (ii) in different fields located in different countries, (iii) containing different growth stages of crop plants and weeds, as well as different weed types and soil conditions. Figure 3.1 illustrates a gallery showing example images that are acquired using different UGVs and UAVs for field tests in this thesis. The images show a large diversity in terms of the appearance of the plants, weeds, and soil conditions. From the collected data, we manually labeled about 26,500 UGV images concerning ground truth information on pixel level. The annotation together

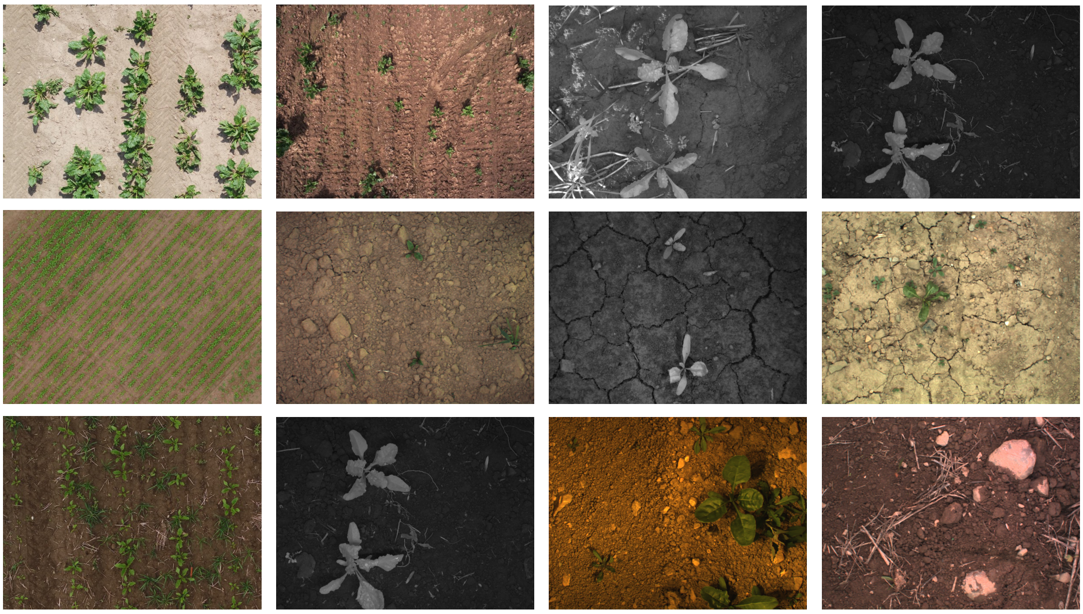


Figure 3.1: Gallery showing example images that are acquired by the employed UGVs and UAVs for field tests. The RGB and NIR images show a large diversity in terms of the appearance of the plants, weeds, and soil conditions. Besides the different field conditions, we recorded the image data with different robots under varying illumination setups.

with the RGB and NIR images form our training and test database (see also Section 3.1.1). In the case of the UAVs, we labeled about 250 high-resolution images (up to 21 megapixels) containing hundreds to thousands of plants each. Note that the UAV images are 10-20 times bigger in size and contain many more plants and weeds compared to the UGV images.

This database is part of our contribution to comprehensive experiments evaluating the performance, generalization capabilities, labeling effort, effect of NIR, and architectural design choices of our proposed approaches along with this thesis. We published parts of our datasets in our dataset paper [20], and as part of a conference paper [82], and journal article [84].

3.1 Robots and Sensor Setup

In this section, we describe the UGV as well as UAV systems used for data acquisition and in-field deployment. For the UGV systems in Section 3.1.1, we describe the basic setup of the field robots, the vision-system, and the actuation system for autonomous weeding. For the UAVs in Section 3.1.1, we describe the used drones and the cameras for image acquisition. As no actuation and no onboard processing is required for the plant classification systems in this thesis, we employ the UAVs solely as flying platforms to transport the camera over the fields. Thus, the focus of this section is more on the acquisition setup.

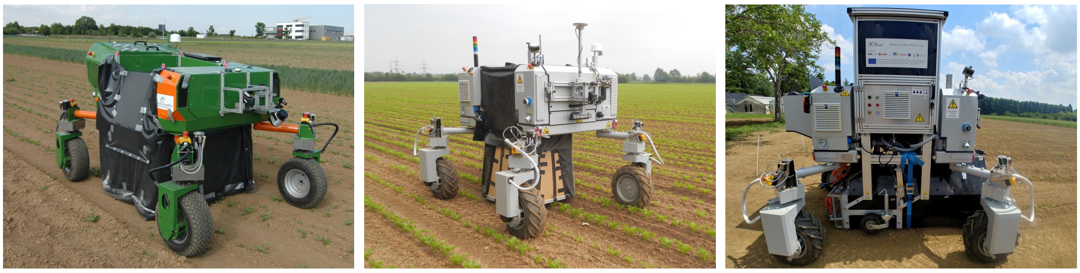


Figure 3.2: Different versions of the BOSCH Deepfield BoniRob used for data acquisition and deployment. Left: BoniRob V2 used for data acquisition in Stuttgart, Germany. Middle: BoniRob V3 used for data acquisition at the Campus Klein Altendorf near Bonn, Germany. Right: BoniRob V3 used in Ancona, Italy.

3.1.1 Unmanned Ground Vehicles

We conduct all field experiments with different generations of the BoniRob field robot, shown in Figure 3.2. The BoniRob is a multi-propose field robot by BOSCH DeepField Robotics and has been developed for agricultural applications such as selective spraying, weed control, as well as plant and soil monitoring. The BoniRob system provides an empty slot to install different tools for specific tasks. In terms of navigation on rough terrain, the robot is equipped with four independently steerable wheels and allows for further flexible movements through a mechanism to adapt its track width to the actual distance between crop rows in the field. In total, the BoniRob is equipped with sensors delivering visual information, depth, 3D laser, GPS data, and wheel odometry measurements. In the remainder of this thesis, we solely rely on the image data as well as the wheel odometry data. Thus, we present the used camera system and do not further explain the other sensors in detail. More information on the other sensors of the BoniRob platform is given in our dataset paper [20].

3.1.1.1 Actuators

The actuation system is responsible for the treatment of the plants in the field. Figure 3.3 shows the weed control unit of the BoniRob, which was developed by BOSCH within the joint EU project Flourish. The weeds are treated either mechanically by bolts or chemically by sprayers. For the high precision mechanical treatment, the unit is equipped with 18 individually controllable stamps with 10 mm-diameter bolts (Figure 3.3 bottom-right). While driving over the field, the controller moves the bolts to the respective positions of the weeds in the object space based on the classification results. Once the bolt is above the shaft area of a weed, it is pneumatically punched into the ground to remove the weed. This mechanical method is most effective with weeds that have a defined stem area. One bolt has a diameter of about 1 cm^2. Therefore, the impact is

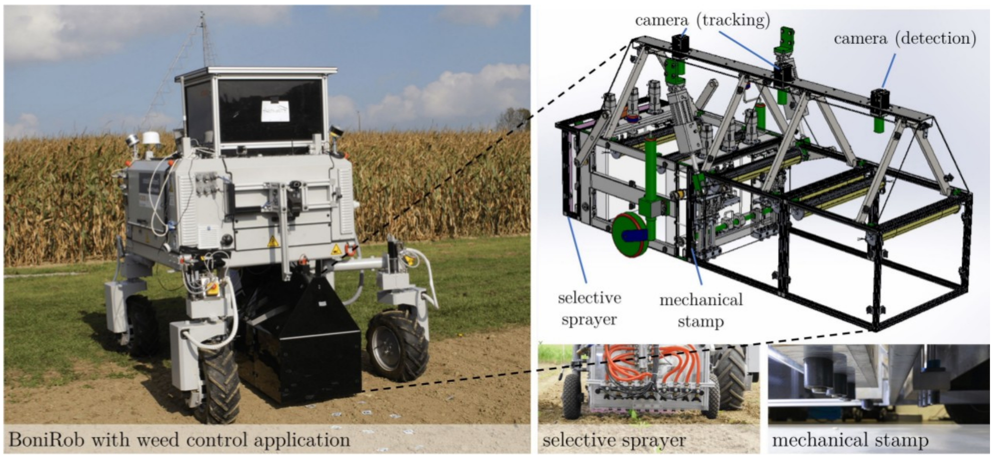


Figure 3.3: Left: BoniRob V3 with weed control unit mounted in the application slot (black box). Right: CAD drawing of weed control unit including the actuators, camera sensors, selective sprayer, and mechanical stamping tool for precision stamping. Bottom row: Examples of sprayed weeds.

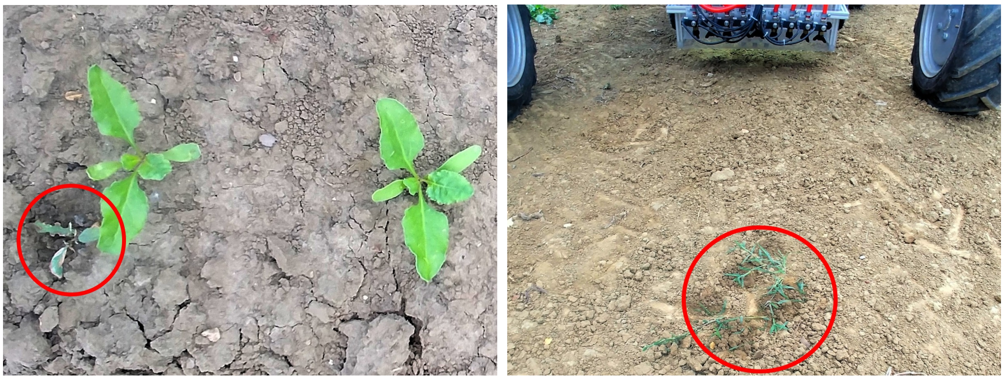


Figure 3.4: Examples of sprayed weeds. BoniRob's spraying system is able to perform the treatment with a spatial precision of around 6 cm. This precision in sufficient to treat also weeds that are located close to crop plants.

most effective on smaller weeds. For larger weeds as well as grass weeds, this approach is not as effective. Here, a chemical treatment is suitable. The advantage of this method is the spatial precision. Theoretically, it can treat weeds with an accuracy of 1 cm^2. Thus, it can treat weeds that grow close to the plants. Conversely, a classification system must determine a reasonably accurate position for the impact location to ensure effective treatments. For this purpose, we develop a classification system in Section 5.3.1, which, in addition to the pixel-wise classification of plants, also detects the exact positions of their stem. In our experiments in Section 6.7.1, we show that we can classify the stem with an average precision of a few millimeters.

For selective spraying, nine individually controllable nozzles are mounted on the back of the unit. The weed control unit is designed for the selective and targeted treatment of plants per row. The width of the spraying array is ap-

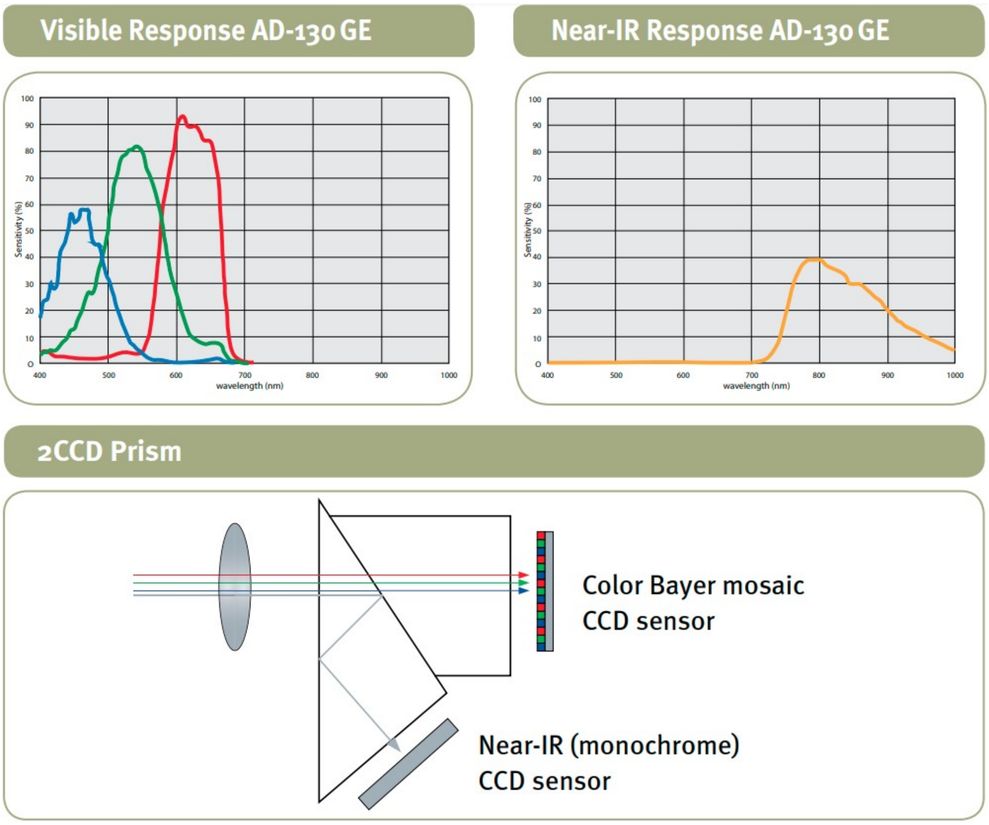


Figure 3.5: Spectral sensitivity of the JAI AD-130 GE camera for the red, green, and blue band as well as for the near-infrared band. One key feature of the JAI camera system is its prism-based design providing two CCD arrays, one CCD for RGB using a Bayer mosaic and another CCD for NIR using a monochrome chip. Image courtesy of [1].

proximately 50 cm. One sprayer covers approximately 6 cm across the driving direction. The sprayers' spatial precision is lower than that of the stamping tool, but the advantage is that the individual nozzles do not move to the location of the plants. Therefore, the spraying system can also operate at higher driving speeds at high throughput. The advantage of spraying is that even large weeds or grass weeds can be treated effectively. Figure 3.4 depicts two examples of sprayed weeds. It is possible to use both actuators simultaneously. For example, small dicotyl weeds can be treated well by mechanical stamping, while larger weeds and grass weeds are better treated chemically by selective spraying.

3.1.1.2 Camera

We used a 4-channel JAI AD-130 GE camera system for image acquisition. The primary purpose of the JAI camera is to capture detailed visual information of the plants for the crop and weed perception system of the robot and detailed visual monitoring of the plant growth by extraction of key indicators for phenotyping applications. As plant leaves exhibit high reflectivity in the NIR spectrum due to

Table 3.1: UAVs used in this thesis.

	Inspire II	Phantom 4	Phantom RTK
Camera	Zenmuse X5s	Built-in	Built-in
Sensor	CMOS	CMOS	CMOS
Sensor size	$\frac{4}{3}''$	$\frac{1}{2.3}''$	$1''$
Focal length	45 mm	3.6 mm	8.8 mm
Resolution	5,280×3,956	4,000×3,000	5,472×3,648

their chlorophyll content [127], the NIR channel is useful for separating vegetation from the soil and other backgrounds in the images.

This camera is a prism-based 2-CCD multi-spectral vision sensor, which provides image data of three bands inside the visual spectrum and observes one band of the near-infrared spectrum. Figure 3.5 illustrates the respective sensitivities of the camera system for the four spectral bands. The Bayer mosaic color CCD and the monochrome CCD of the JAI camera provide an image resolution of $1,296 \times 966$ pixels, respectively. One key feature of this camera system is its prism-based design: as the optical paths of the RGB and the NIR channel are identical, the RGB and NIR data can be treated as one 4-channel image.

The primary purpose of the JAI AD-130GE camera is to capture detailed visual information of the plants for the crop and weed perception system of the robot and detailed visual monitoring of the plant growth by extraction of key indicators for phenotyping applications. As plant leaves exhibit high reflectivity in the NIR spectrum due to their chlorophyll content [127], the NIR channel is useful for separating vegetation from the soil and other backgrounds in the images.

The camera points downwards on the field approximately 60 cm-85 cm above the soil. See Figure 3.5 (right) for an illustration of the camera mounted in the weed unit module. We use a Fujinon TF15-DA-8 lens with a fixed focal length of 8 mm leading to a field of view of 25 cm-35 cm in driving direction and 32 cm-45 cm orthogonal to it. This setup yields a ground sampling distance of around $0.3\frac{mm}{px}$-$0.4\frac{mm}{px}$.

3.1.2 Unmanned Aerial Vehicles

We carry out aerial data acquisition and field experiments with different UAVs. The goal of this work is to find out if and with which quality we can derive the spatial distribution of plants and weeds, or even different weed species from RGB aerial image data. In this work, we use the UAVs exclusively as mobile platforms to transport the camera sensors over the field. This enables us to observe large areas quickly and to analyze them with our classification systems in a comparatively short time. Note that we do not perform any on-board computing on the UAV. Figure 3.6 shows the drones we

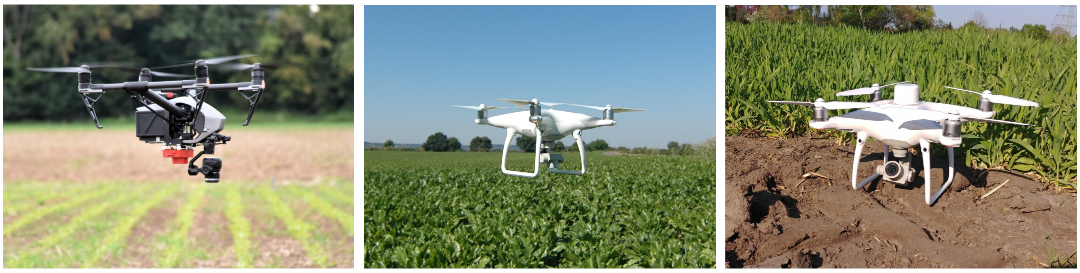


Figure 3.6: Different UAVs used for data acquisition. Left: DJI Phantom 4 with built-in RGB camera. Middle: DJI Inspire II with DJI Zenmuse X5s RGB camera. Right: DJI Phantom 4 RTK with built-in RGB camera

use to capture our UAV records, which we describe in Section 3.2.3 and Section 3.2.4. We use various UAV systems from the manufacturer DJI. Compared to the camera we use for the UGVs, the UAV camera systems are always standard RGB cameras. The crucial differences to the UGV acquisition conditions are that the resolution of the cameras is much higher and that the UAV images are not acquired under artificial lighting. Therefore, the data can vary considerably, even if they are collected within a short time by a single flight over a single field. While the UGV camera provides a resolution of around one megapixel, the UAV cameras provide images in the range of 13 to 21 megapixels.

3.2 Datasets

In order to develop and evaluate the proposed plant classification systems, we gathered a substantial amount of data from different fields located in different cities and countries such as Bonn, Germany, Ancona, Italy, Stuttgart, Germany, and Zurich, Switzerland. We collected these datasets with the UGVs and UAVs we described in the previous section. Overall, these datasets represent challenging conditions for a vision-based classification system. They contain sugar beet plants at different growth stages, which we consider as the value crop, different dicotyl weed (weeds whose seeds having two embryonic leaves), and grass weed types at varying sizes as well as different soil conditions. Even within a single dataset recorded during a single run with the robot in one field, the size of the crops and weeds can range between $1\,\text{cm}^2$-$20\,\text{cm}^2$.

This variation is caused by natural differences in the growth of plants and weeds but also due to the way the images are labeled. Note that we do not determine the crop and weed sizes based on the total pixels of a plant but through a connected component analysis on the label map. Therefore, the measured object or segment sizes are sometimes smaller, as plants are not always represented by a single connected component in the image space. The label map for BONN-CW-17 in Figure 3.7 illustrates the latter situation. Here, distinct connected components represent the leaves of the sugar beet plant. Furthermore, the image data of the respective datasets differ also in color, brightness, and contrast, due to the changing illumination setups of the field robots.

We categorize the acquired datasets into four main groups, i.e.:

1. In Section 3.2.1, we describe the crop-weed classification datasets acquired by UGVs. These datasets represent the main source of data in this thesis with around 20,000 labeled images containing the classes sugar beet as our considered crop ω_{c}, weed ω_{w}, and soil background ω_s.

2. In Section 3.2.2, we describe our crop-dicot-grass classification and stem detection datasets for the UGV acquired by UGVs. These datasets represent the data source to develop and evaluate our joint plant classification and stem detection systems for selective and species-specific in-field treatments. They consist of around 5,000 labeled images considering the classes: sugar beet ω_{c}, dicotyl weed ω_d, grass weed ω_g, and soil background ω_s for the plant classification. Furthermore, we labeled the stem locations for the sugar beets ω_{c} and dicotyl weeds ω_d.

3. In Section 3.2.3, we describe the plant classification datasets acquired by UAVs. Here, we distinguish two types of datasets. First, single non-overlapping images with a high spatial ground resolution of around 1 mm. These datasets contain either the classes sugar beet ω_{c}, weed ω_{w}, and soil background ω_s, but also images labeled for multiple weed species detection considering four different types of weed. The second dataset type contains overlapping images with a ground resolution of about 5 mm. We first process the images with the photogrammetric software Metashape to obtain a single Orthomosaic, which is a true-scale representation of stitched orthophotos obtained by a bundle-adjustment procedure.

4. In Section 3.2.4, we present our crop counting dataset acquired by UAVs. This dataset contains the measurements of a single field acquired at different points in time. We labeled the data in terms of a pixel-wise classification of sugar beet ω_{c}, weed ω_{w}, and soil background ω_s, but also terms of the number of present crops in the data.

3.2.1 UGV Crop-Weed Datasets for Plant Classification

First, we describe our UGV-based crop-weed datasets for plant classification. During the EC-funded Flourish project [120], we used different versions of the BoniRob platform for data collection and field testing. In total, we conducted experiments on five different fields in three different countries in central Europe. We collected a diverse database, allowing us to evaluate our classification systems under different real-world conditions.

Throughout this section, we introduce five different datasets for the plant classification, considering the classes crop ω_{c}, weed ω_{w}, and soil background ω_s. We first introduce a uniform notation concerning the datasets present below. We refer to these datasets with CITY-CWS-YEAR, where CITY stands for the city in which we recorded the data. The term CWS refers to crop, weed, and soil. The term YEAR refers to

Table 3.2: Key statistics of the UGV crop-weed datasets.

Dataset	**Images** [#]	**Pixels** [%]				**Objects** [#]	
		including soil		excluding soil			
		(ω_c, ω_w)		(ω_c, ω_w)		(ω_c, ω_w)	
BONN-CW-16	12,429	1.3	0.3	80	20	36,103	68,190
BONN-CW-17	1,854	1.1	0.8	57	43	2,695	4,272
STUTT-CW-15	3,462	1.5	0.6	69	31	6,654	20,439
ANCONA-CW-18	1,214	0.2	0.6	32	68	641	4,880
ZURICH-CW-16	2,578	0.1	0.5	21	79	5,257	36,501

the year in which we recorded the data. In the remainder of this section, we present our datasets BONN-CW-16, BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16. Figure 3.7 depicts an example RGB and NIR image pair and its corresponding ground truth for each of the datasets.

For the entire UGV-based data acquisition, the camera field of view was shaded to be as independent as possible from the natural light source. The artificial light setup inside the shaded area, however, changes for the different versions of the BoniRob. In the initial version of the BoniRob V2, the artificial lighting is provided by a series of halogen bulbs, which resulted in a "spotty" illumination of the scene. In contrast, for the BoniRob V3 versions, a LED-tube based system consisting of diodes in the red, green, blue, and infra-red spectrum was used, which provided a more uniform illumination. We collected the STUTT-CW-15 dataset with the BoniRob V2 and the other dataset with BoniRob V3.

BONN-CW-16: This dataset represents our main source of training data for plant classification systems in this thesis and is published in our dataset paper [20]. In spring 2016, we started to conduct a two-month data acquisition campaign at Campus Klein Altendorf near Bonn in Germany. We collected data on a sugar beet field during a crop season, covering the various growth stages of the plants, see Figure 3.9. On average, we acquired data on two to three days a week, leading to 30 days of recordings in total. Figure 3.8 depicts the trajectories of the GPS sensor of the BoniRob. The different colors of the tracks refer to different dates of acquisition. We used the BoniRob V3 platform that is depicted in Figure 3.2 (middle). On a typical day's recording, the robot covers between four to eight crop rows, each measuring 400 m in length. We intended to capture the key variations of the field during the time relevant for weed control and crop management. Thus, the data collection process was phased over-time to cover the different growth stages of the sugar beet crop starting at germination. The dataset also captures different weather and soil conditions ranging from sunny and dry to overcast and wet. In comparison to the other crop-weed datasets in this section, it is the largest one. We manually labeled around 12,500 RGB and NIR image pairs. This dataset serves a challenging classification task, as classifiers need to consider a large variety of growth stages, several weed types, and different soil conditions caused by changing weather conditions during the data acquisition throughout the entire season.

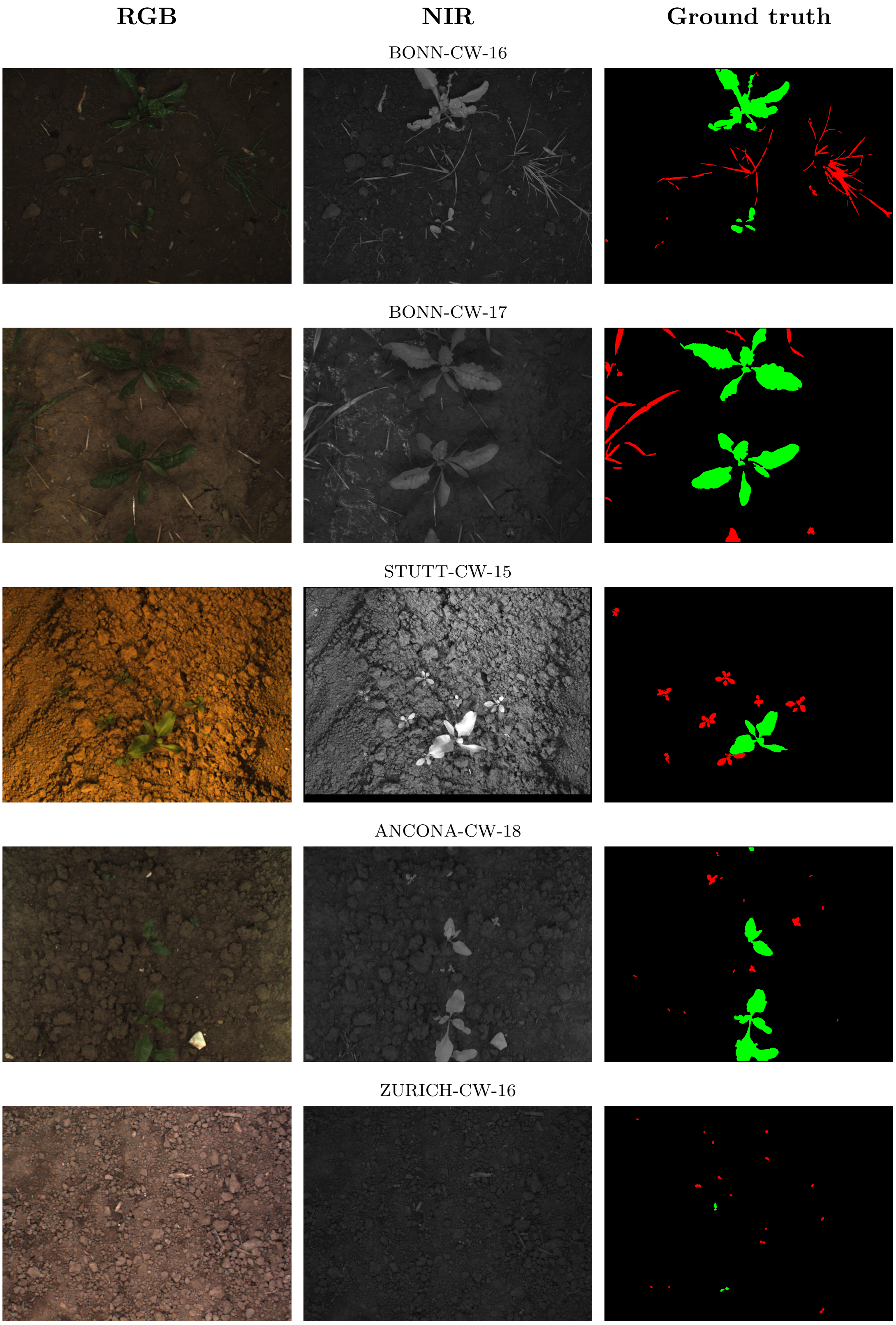


Figure 3.7: Example RGB and NIR images and corresponding pixel-wise ground truth information for the plant classification considering the classes crop ω_c (green), weed ω_w (red), and soil background ω_s (black). The datasets differ from each other in terms of environmental changes, including varying weed pressure, various weed types, different growth stages of crop plants and weeds, and illumination conditions. We show further examples in Chapter 6.

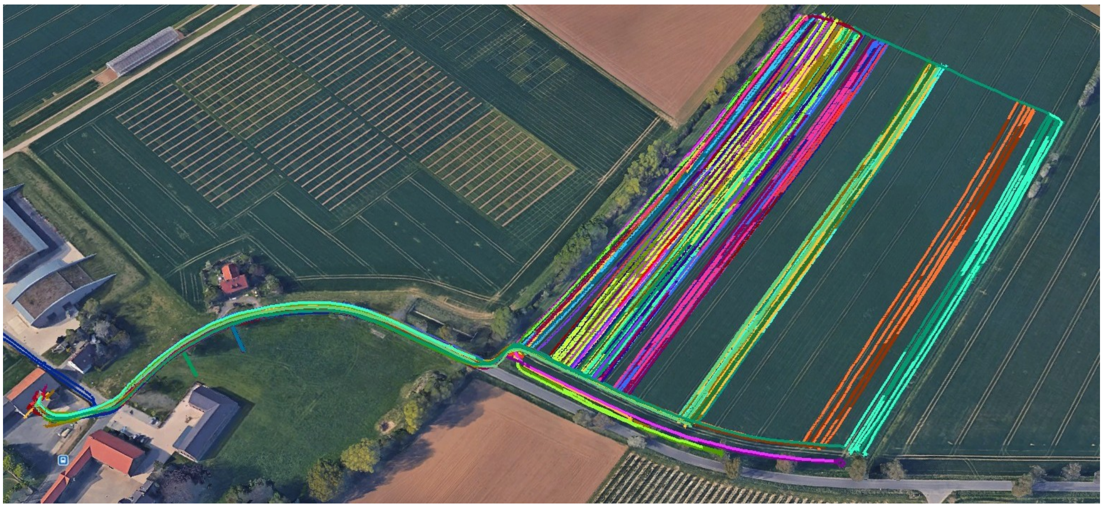


Figure 3.8: Paths estimated by the GPS sensor of the entire data acquisition campaign at the Campus Klein Altendorf. Different colors refer to recordings of different days. Best viewed in color.

BONN-CW-17: In autumn 2017, we recorded this dataset as well at the Campus Klein Altendorf. In contrast to the BONN-CW-16 dataset, we used the BoniRob V3 platform that is depicted in Figure 3.2 (right). The dataset contains around 1850 labeled RGB and NIR image pairs containing crops in a 2-4 leaves stage and big crops in a 6-12 leaves stage. The size of the sugar beets ranges from $0.5\,\mathrm{cm}^2$-$20\,\mathrm{cm}^2$. A similar variation is also present in the size distribution of weeds, i.e., $0.5\,\mathrm{cm}^2$-$15\,\mathrm{cm}^2$. We collected the data by driving over two distinct areas in the field. In both areas, the plants were sown at different points in time.

STUTT-CW-15: In spring 2015, we recorded this dataset in cooperation with BOSCH DeepField Robotics near Stuttgart, Germany. It contains around 3,500 labeled RGB and NIR image pairs. We used the BoniRob V2 depicted in Figure 3.2 (left) to acquire data on three days. On two days, we recorded data in the same area of the field with a temporal difference of a week. We performed the third recording a few weeks later than the first two recordings, but on a distinct area in the same field. Thus, the soil conditions in this dataset are mostly comparable, but the dataset contains a substantial amount of differently sized crop plants and weeds both ranging from $1\,\mathrm{cm}^2$ to $25\,\mathrm{cm}^2$. The artificial illumination conditions induce a large difference to the other datasets. We use a series of halogen bulbs, which result in a “spotty” illumination of the scene, whereas in the other datasets, we use LED-tubes, providing more uniform illumination. Furthermore, this dataset contains a substantial amount of weeds that overlap with the crop plants. Thus, the dataset represents challenging conditions for plant classification systems. In our experiment in Section 6.3.1.2, we split a subset STUTT-CW-15-SUB from this dataset consisting of 1,718 images that mostly represent these conditions, see also Figure 3.10.

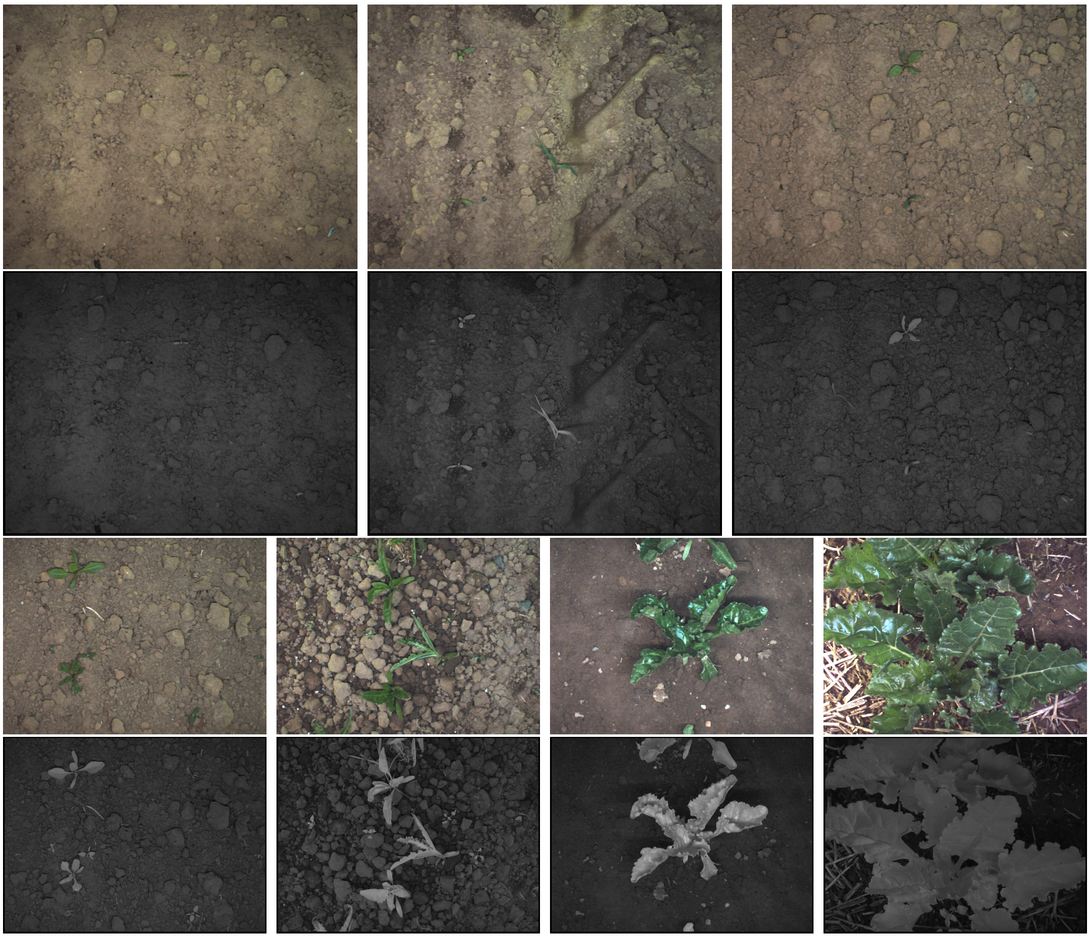


Figure 3.9: Sugar beets and weeds of the BONN-CW-16 dataset captured with the JAI AD-130GE multi-spectral camera. The first and third rows show RGB images. The rows below show the corresponding NIR images. The image data in the BONN-CW-16 dataset contains sugar beet data from its emergence (first/second row) up to the growth stage at which machines are no longer used for weed control (third/fourth row).

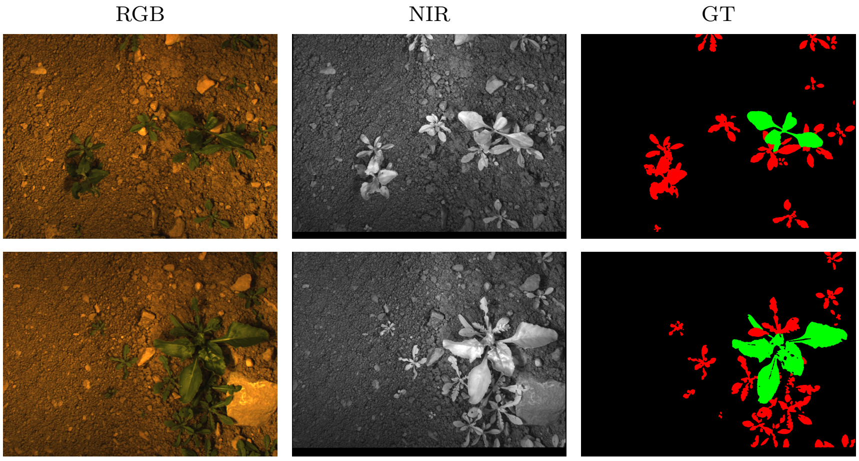


Figure 3.10: Example RGB+NIR images from the STUTT-CW-15 and STUTT-CW-15-SUB dataset, corresponding ground truth (GT) information with sugar beet ω_c (green), weed ω_w (red), and soil background ω_s (black).

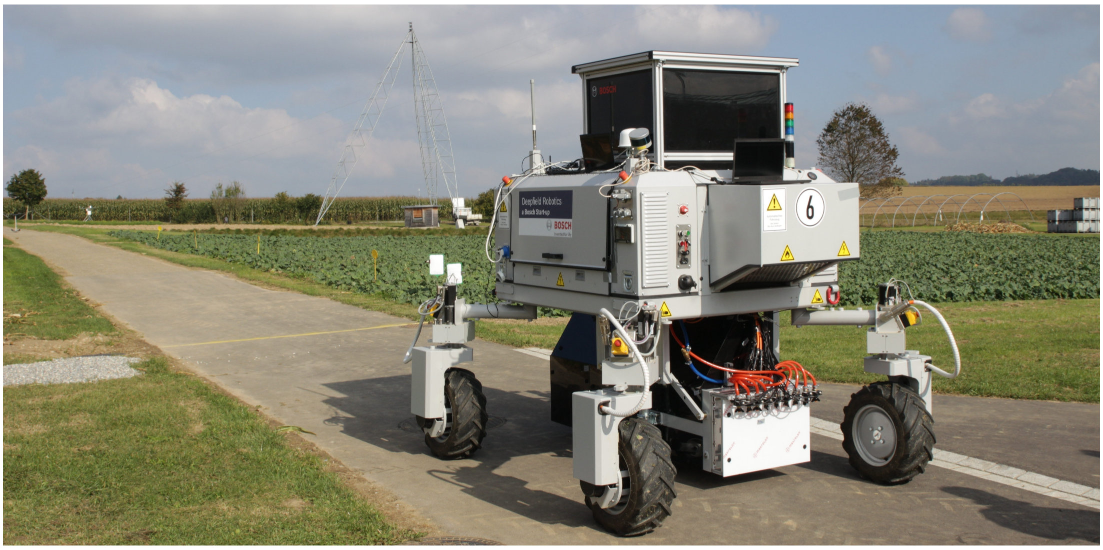


Figure 3.11: BoniRob operating at fields from the Field Phenotyping Platform at the Eschikon Field Station of ETH Zurich [64].

ANCONA-CW-18: In spring 2018, we recorded this dataset near Ancona, Italy. We used the BoniRob V3 platform that is depicted in Figure 3.2 (right). The dataset contains around 1,200 labeled RGB and NIR image pairs containing both small crops in a 2-4 leaves stage and big crops in a 6-12 leaves stage. As for the BONN-CW-17 dataset, we collected the data by driving over two distinct areas in the field. In both areas, the plants were sown at different points in time. This dataset holds a substantial amount of weeds. Almost 70 % of the vegetation pixels belong to the weed class. The datasets contain different weed species of different sizes ranging from 0.1 cm^2-15 cm^2. It contains both dicotyl as well as grass weeds.

ZURICH-CW-16: In autumn 2016, we recorded this dataset at the Field Phenotyping Platform (FIP [64], see Figure 3.11) at the Eschikon Field Station of Eidgenössische Technische Hochschule Zurich, Switzerland. We used the BoniRob V3 platform. We collected the images on three different dates once a week. As the temperatures in autumn in Zurich were comparably cold compared to the temperatures that are necessary for the normal growth of the plants, throughout all measurements, the plants are mostly in a typical growth stage as right after emergence, i.e., two leaves unrolled and more leaves not viable. Thus, ZURICH-CW-16 is the most challenging dataset as it contains mostly small crop plants and weeds right after their emergence phase in the field. Around 90 % of the weeds and 66 % of the crop plants have a size between 0.1 cm^2-0.8 cm^2 and thus are only represented by a few pixels. Furthermore, crop plants and weeds have a similar appearance at this growth stage.

Cross-Dataset Domain Shift

For our crop-weed datasets, we further analyze how different the respective datasets are to each other concerning the visual appearance of the images. Through the thesis, we

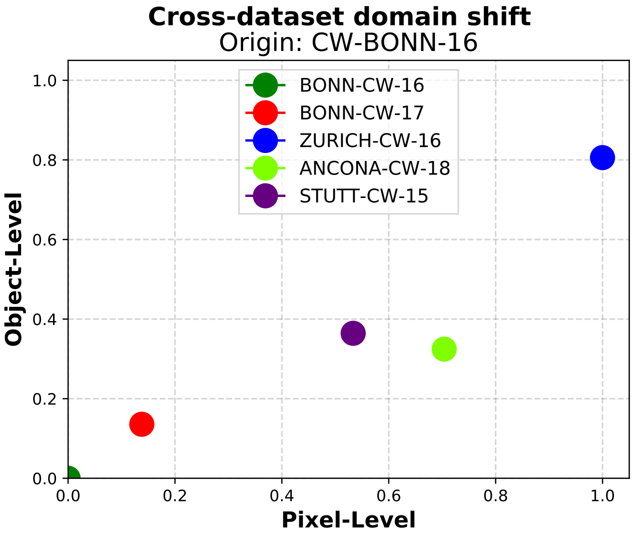


Figure 3.12: Domain shifts between the BONN-CW-16 dataset and all other crop-weed datasets.

call this difference the domain shift between datasets. The domain shift between two datasets acquired in different field environments exists due to different soil conditions, growth stages of the plants and weeds, different weeds types, but also due to different illumination setups for the camera system.

The goal in this section is to quantify the domain shift that the classifiers have to overcome when being trained on a single dataset, i.e., a particular field environment, and then being deployed in another field. Understanding the domain shift helps to better understand the classifiers' performance, especially in terms of their generalization capabilities to new field environments. The higher the domain shift between training and a test dataset is, the better a classifier has to generalize to achieve a particular performance.

We define a set of dataset-specific properties on pixel- and object-level. Pixel-level properties describe the class-wise distribution of the RGB and NIR intensities as well as the probability of occurrence of individual classes in the data. The object-level properties describe the probability of class-wise objects, i.e., connected components in the label space and the distribution of the size of objects. Note that these properties are based on the ground truth information. Thus, they cannot be considered through an unsupervised preprocessing procedure.

First, we preprocess all images using our proposed preprocessing procedure in Section 4.2. The preprocessing mainly performs a color correction and a contrast enhancement of the images. On pixel-level, we compute the respective means and standard deviations for the red, green, blue, and nir channel for the crop, weed, and soil class. Furthermore, we compute the ratio of the crop, weed, and soil pixels. The latter, we report in Table 3.2. Through this, we obtain 27 features describing the domain of a dataset on pixel-level in a class-wise manner. At the object-level, we compute the mean and standard deviation of the sizes of the class-wise objects and the relative frequency of the class-wise occurrence. Thus, we obtain 6 features describing the domain of a dataset at the object-level.

To quantify the domain shift between two datasets, we compute the Euclidean distance between the respective features. Before we compute the Euclidean distances, we normalize each feature to be in the range of $[0, 1]$ concerning its minimum and maximum value across all datasets. Figure 3.12 illustrates the resulting domain shifts between the BONN-CW-16 dataset and all other datasets in a two-dimensional coordinate system. The x-axis refers to the object-level properties, and the y-axis refers to the pixel-level properties. Note, we directly illustrate in Figure 3.12 the domain shift of each dataset concerning the BONN-CW-16 dataset. Therefore, the BONN-CW-16 dataset is at location $(0, 0)$, as it is equal to itself. We choose this illustration, as we perform most of the crop-weed classification experiments in our experimental evaluation by training our classifiers on the BONN-CW-16 dataset and deploying them on the other dataset.

We see that the ZURICH-CW-16 dataset has the most considerable domain shift for both pixel-level and object-level. Figure 3.7 illustrates the difference in color but also in terms of plant size. The STUTT-CW-15 and the ANCONA-CW-18 datasets are similar in terms of object-level but have a lager domain shift at the pixel-level. The BONN-CW-17 dataset is most similar to the BONN-CW-16 dataset.

3.2.2 UGV Crop-Dicot-Grass Datasets for Plant Classification and Stem Detection

Next, we describe our UGV-based crop-dicot-grass (CDGS) datasets for plant classification and stem detection. In order to evaluate the pixel-wise classification and the plant stem detection performance as well as the generalization capabilities of the classification system to unseen fields, we gathered data from different fields located in different cities in different countries such as (1) Bonn, Germany, (2) Ancona, Italy, (3), Stuttgart, Germany, and (4) Zurich, Switzerland. Our notation for referring to the respective datasets is the same as for the crop-weed datasets presented in the previous section. Note that the BONN-CDGS-16, ANCONA-CDGS-18, and STUTT-CDGS-15 dataset are from the same field as in the case of the crop-weed datasets, whereas the ZURICH-CDGS-17 dataset is from another field than the Zurich dataset from the previous section.

All datasets contain sugar beet plants (crop) and dicotyl weeds. BONN-CDGS-16 and ANCONA-CDGS-18 contain a substantial amount of grass weeds. No grass weeds are present in the STUTT-CDGS-15 dataset, and only a minimal number of grasses are present in the ZURICH-CDGS-17 dataset. In addition to the pixel-wise labels for the plant classification task, we also label the stem locations, i.e., a particular pixel location for the crop plants and dicotyl weeds. The datasets represent challenging conditions for a vision-based classification system, as they contain different dicotyl weed and grass weed types with varying sizes and different soil conditions. The image data differs in color, brightness, and contrast due to changing light setups of the field robots. Figure 3.13 shows examples from each dataset for the crop-dicot-grass classification and stem detection. Table 3.3 summarizes the key statistics for each dataset.

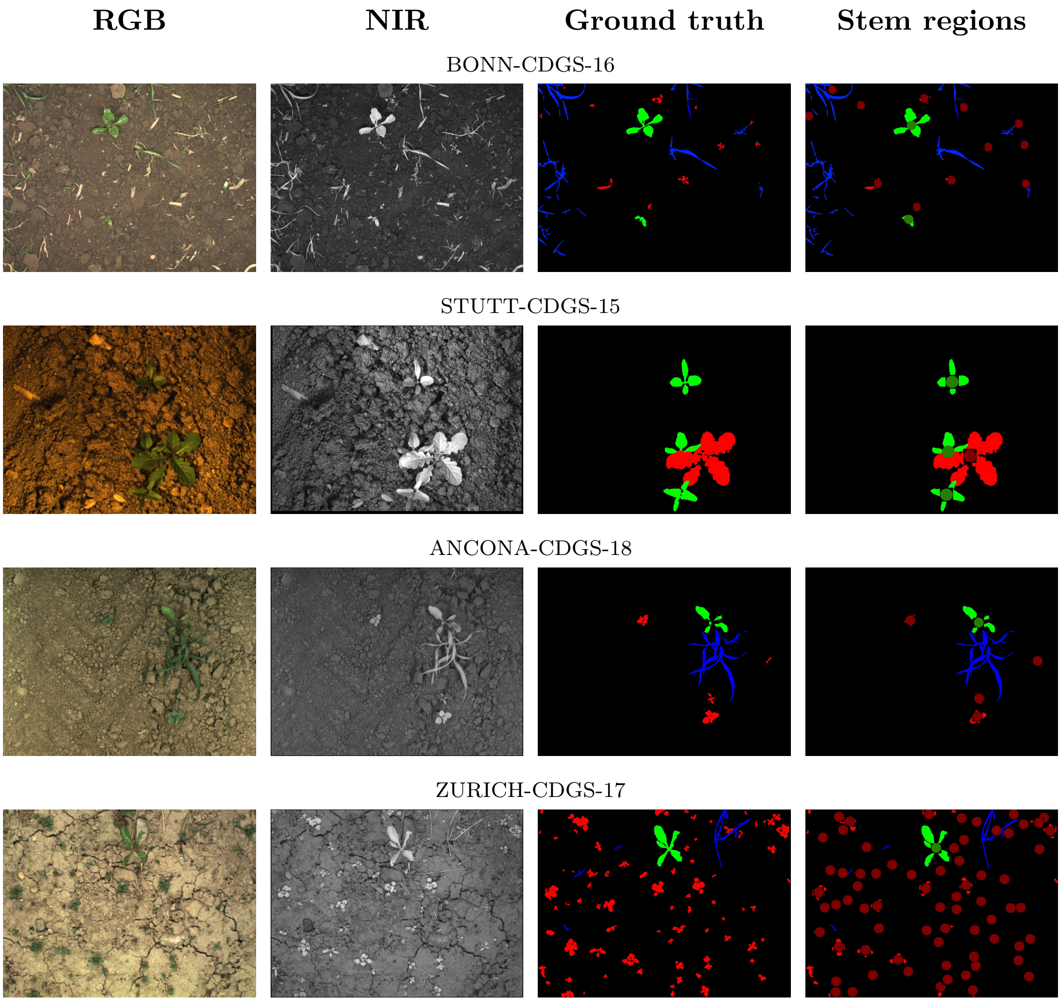


Figure 3.13: Example RGB and NIR images and corresponding pixel-wise ground truth information (i) for the pixel-wise classification of sugar beet ω_{c} (green), dicotyl weed ω_{w} (red), grass weed ω_{w} (blue), and soil background ω_s (black) and (ii) for the stem detection task. The datasets differ from each other in terms of environmental changes, including varying weed pressure, various weed types, different growth stages of crop plants and weeds, and illumination conditions. We show further examples within our experimental Chapter 6.

Table 3.3: Key statistics of the UGV crop-dicot-grass datasets.

Dataset	**Images** [#]	**Pixels** [%]			**Objects** [#]			**Stems** [#]	
		excluding soil							
		$(\omega_{\mathrm{c}}, \omega_d, \omega_g)$			$(\omega_{\mathrm{c}}, \omega_d, \omega_g)$			$(\omega_{\mathrm{c}}, \omega_d)$	
BONN-CDGS-16	2,291	69	19	11	4158	28,739	7,261	3,083	19,911
STUTT-CDGS-15	2,086	71	28	1	3,126	4,776	53	2,476	4,218
ANCONA-CDGS-18	284	67	13	21	797	914	213	542	1,019
ZURICH-CDGS-17	62	55	44	2	56	1,039	27	41	1,130

BONN-CDGS-16: With almost 2,300 labeled RGB and NIR image pairs, it contains the most samples of our crop-dicot-grass datasets and holds a substantial number of plants from all the considered classes, i.e., crop, dicotyl weeds, and grass weeds. The images from the BONN-CDGS-16 dataset are a subset of our published dataset paper [20]. We selected those parts of the BONN-CW-16 dataset that contain a substantial amount of grass weeds in order to have a sufficient amount of training examples for this class.

STUTT-CDGS-15: This dataset consists of images of the STUTT-CW-15 dataset, which were recorded on one of the measurement days. The dataset does not contain grasses, but we labeled almost 2,100 RGB and NIR image pairs of weed stems and dicotyl weeds to have another dataset for stem detection from another field for the evaluation of generalization properties to new and changing field properties. The dataset holds around 2,500 crop stems and 4,200 dicotyl weed stems.

ANCONA-CDGS-18: This dataset consists of 284 labeled RGB and NIR image pairs from the ANCONA-CW-18 dataset. With 67 % crop pixels, 13 % weed pixels, and 21 % grass pixels, proportional to the total vegetation pixels, the ANCONA-CDGS-18 represents a balanced dataset concerning the distribution of the individual classes. With 284 images, it is way smaller compared to BONN-CDGS-16 and STUTT-CDGS-15. On the other hand, it serves a substantial amount of grass weeds and is appropriate for analyzing the classification performance of this class.

ZURICH-CDGS-17: In autumn 2017, we recorded this dataset at the FIP in Eschikon near Zurich on the same field as for the ZURICH-CW-16 data. However, after a difference of almost one year, it contains different soil conditions. Furthermore, the crop plants, dicotyl weeds, and grass weeds were recorded in a different growth stage concerning the ZURICH-CW-16 data. Thus, we see no relation between the ZURICH-CDGS-17 and ZURICH-CW-16 dataset, except that the considered crop is sugar beet. We used the BoniRob V3 platform that is depicted in Figure 3.2 (right). We collected the images during a single run over the field. The dataset consists of 68 labeled RGB and NIR image pairs. The crops are within a 4-8-leaf growth stage. For the data recording, we selected a region in the field which was not treated with chemicals for weed control. Thus, the data contains substantial weed pressure, see Figure 3.13. In numbers, we labeled 28 dicotyl weeds for one sugar beet leading to 41 sugar beets and 1,130 dicotyl weed stems.

3.2.3 UAV Crop-Weed Datasets for Plant Classification

Next, we describe our UAV-based crop-weed datasets for plant classification in aerial imagery. In order to evaluate the pixel-wise plant classification performance as well as the generalization capabilities of the classification systems to new and changing field conditions, we gathered data from two different fields located in Bonn, Germany, and

Table 3.4: Key statistics of the UAV crop-weed datasets.

Dataset	**Images** [#]	**Pixels** [%]				**Objects** [#]	
		including soil		excluding soil			
		$(\omega_\mathsf{c}, \omega_\mathsf{w})$		$(\omega_\mathsf{c}, \omega_\mathsf{w})$		$(\omega_\mathsf{c}, \omega_\mathsf{w})$	
BONN-UAV-17-1MM	94	9	3	76	24	5.525	17,914
ZURICH-UAV-17-1MM	88	9	4	67	33	3,660	18,263
BONN-UAV-17-5MM	20	16	5	72	28	2,123	3,850

Zurich, Switzerland. Our notation for referring to the respective datasets is similar as for UGV-based the crop-weed datasets. The name of a dataset consists of the city, the term UAV, and the year of its acquisition.

Throughout this section, we propose datasets for the UAV-based plant classification considering the classes crop ω_c, weed ω_w, and soil background ω_s. First, we present the BONN-UAV-17-1MM and ZURICH-UAV-17-1MM high-resolution datasets with a ground sampling distance of about 1 mm per pixel. Table 3.4 summarizes the key statistics for these datasets. Both datasets consist of around 90 pixel-wise annotated images containing sugar beets and weeds observed in different growth stages and under different weather conditions. Furthermore, we introduce the BONN-UAV-M-16 dataset, which we describe in Section 3.2.3. The dataset consists of 20 fully annotated images containing sugar beets and several weeds species that we manually labeled considering sugar beets, saltbush, chamomile, other weeds, and soil. Third, we describe our BONN-UAV-17-5MM dataset. This dataset is used to evaluate the plant classification at a comparatively low soil resolution of about 5 mm. As the classifiers have to differentiate between crop plants and weeds based on less pixel information, the lower resolution represents more challenging conditions for the classifiers. However, the lower resolution leads to a large spatial throughput of a single flight as the observed area increases to the square of the ground sampling distance.

BONN-UAV-17-1MM: In autumn 2017, we recorded this dataset near Bonn in Germany at the Campus Klein Altendorf. The images of this record are taken on the same field as the UGV dataset BONN-CW-17. The dataset consists of 94 RGB aerial images, which we collected in five flights within three weeks. We used the INSPIRE II for the acquisition and obtained images with a resolution of 5,280×3,956 pixels. The flight took place at an altitude of 15 m. With the given camera setup, which we describe in Section 3.1.2, this leads to a ground resolution of about 1 mm per pixel. The field has several areas where the plants were sown at different times. Therefore, the dataset includes sugar beets distributed across all growth stages. Besides, we made sure that weeds could also develop. The weather was different during data acquisition. On some days it was rainy and on some days sunny. Figure 3.14 shows two exemplary RGB images with the respective ground truth information. Already in the early growth phase, we see that there are numerous weeds in the field. At a later stage, the high weed pressure creates challenging conditions for the classifiers. The weeds spread unhindered

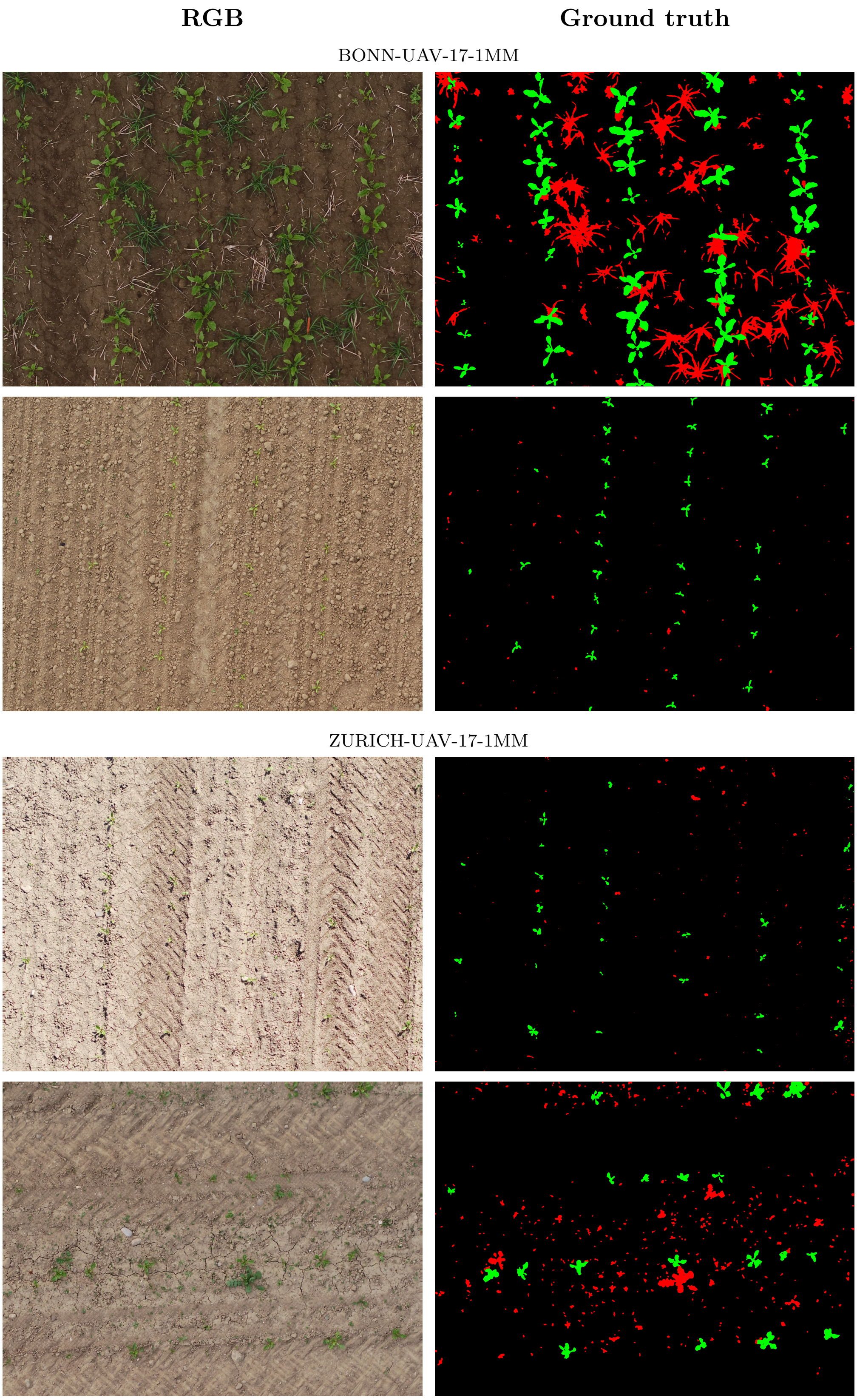


Figure 3.14: Example UAV images and corresponding pixel-wise ground truth information for the plant classification considering the classes crop ω_{c} (green), weed ω_{w} (red), and soil background ω_s (black).

and thus overlap with the plants between and within the row. Also, when looking at the images, we can further see that the crops grow in rows and are of similar distance between and within the row.

ZURICH-UAV-17-1MM: In autumn 2017, we recorded this dataset at the FIP in Eschikon near Zurich, Switzerland. The images of this record are taken on the same field as the UGV dataset ZURICH-CDGS-17. We recorded this dataset in cooperation with the Institute of Agricultural Sciences of the Eidgenössische Technische Hochschule in Zurich. The dataset consists of 84 labeled RGB aerial images, which we have taken over four weeks. For the image acquisition, we used the Inspire II with the same flight and camera setup as for the BONN-UAV-17-1MM dataset. Also, for this dataset, we did not carry out any weed control so that the weeds could develop unhindered in the field. The weather was very sunny during the whole time of data acquisition. As a result, the exposure conditions differ from the BONN-UAV-17-1MM dataset. In contrast to the BONN-UAV-17-1MM dataset, soil conditions were not particularly good for plant development, and animals ate some of the plants shortly after emergence. Figure 3.14 shows two sample images for the ZURICH-UAV-17-1MM dataset. We see that there is a high weed pressure in both the early and late growth stages of sugar beet. We also see that some of the sugar beets are missing in the data. This means that the row information is not as stable as for the BONN-UAV-17-1MM dataset.

BONN-UAV-M-16: In summer 2016, we recorded this dataset near Bonn in Germany at the Campus Klein Altendorf. The data is captured in a border region of a sugar beet field where the plants are not sowed in crop rows. The dataset provides images obtained by an unmodified consumer DJI Phantom 4 UAV. The obtained ground resolution of $0.8\,\frac{mm}{px}$ is comparably high. The images were captured with a resolution of $4,000 \times 3,000$ pixels at a flight altitude of 3 m-4 m. Due to the resolution, we can visually identify typical weeds in sugar beet fields, i.e., saltbush as a common problem weed in terms of mass, chamomile, and other weeds. We fully labeled 20 images of this dataset and show two examples in Figure 3.15. The dataset consists of 20 fully annotated images containing sugar beets and several weeds species that we manually labeled considering sugar beets, saltbush, chamomile, other weeds, and soil. We partially used this dataset in our publication [88].

BONN-UAV-17-5MM: In autumn 2017, we recorded this dataset near Bonn in Germany at the Campus Klein Altendorf. The images of this record are taken on the same field as the BONN-UAV-17-1MM dataset. We used the Inspire II drone for the acquisition.

There are two major differences to the high-resolution BONN-UAV-17-1MM dataset. First, the ground resolution is five times lower at 5 mm per pixel. Second, this dataset does not consist of several single images, but a georeferenced orthomosaic. The orthomosaic, sometimes also called true orthophoto, is a geometrically corrected orthorectified image with a uniform scale and ground sampling distance. Thus, it is corrected

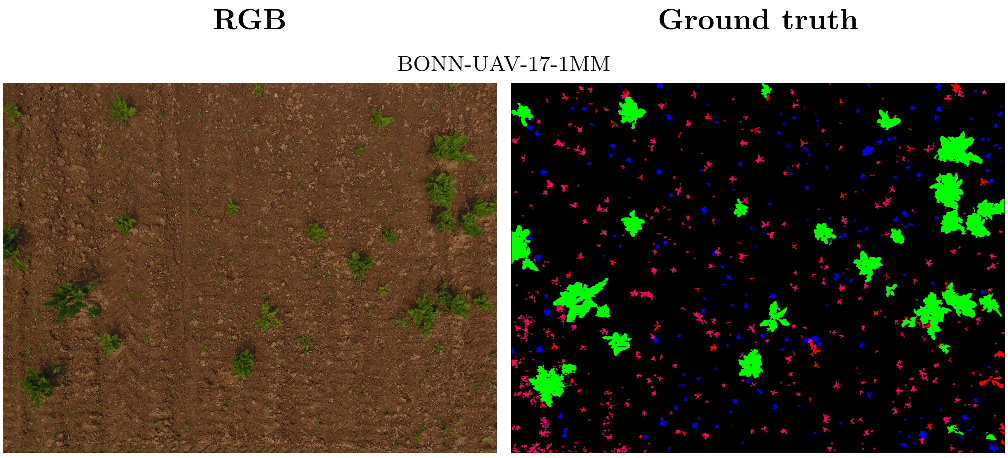


Figure 3.15: Example UAV images of the BONN-UAV-M-16 dataset and corresponding pixel-wise ground truth information for the plant classification considering the classes crop in green, saltbush in blue, chamomile in magenta, other weed in red, and soil background.

due to perspective distortions by using a digital surface model. The scale of the orthomosaic is preserved, and therefore the orthomosaic can be used for measurements. We compute the orthomosaic by feeding the images from the flight into Metashape, a photogrammetric software to build 3D surfaces, orthomosaics, and digital elevation models. In other words, the orthomosaic is an image representation of an area that is created from several images that are stitched together and geometrically corrected.

We acquired 250 images by flying the Inspire II drone at around 75 m altitude over the entire field. We planned a regular grid as a flight route with a photogrammetric flight planning tool. We set the desired image overlap to 60 % along and perpendicular to the flight direction. Figure 3.16 illustrates the entire north-oriented orthomosaic of the BONN-UAV-17-5MM dataset. The red bounding box defines the area in which we conduct the crop-weed classification experiments. The zoomed views show sugar beet plants and weeds and reveal that the sugar beet plants have different sizes and colors as well as weeds growing between an in the crop. For the evaluation of this data, we extracted 20 patches (blue boxes) from the orthomosaic for which we provide pixel-wise labeling into the classes crop, weed, and soil.

3.2.4 UAV Crop-Weed Datasets for Plant Counting Under Harsh Conditions

Finally, we describe our UAV-based crop-weed datasets for plant counting in harsh field conditions. The goal of this dataset is to evaluate our approaches to automatically provide exact knowledge about the number of emerged plants, as this information reflects an essential trait for both farmers and breeders. Harsh conditions in this context mean that the classifier has to deal with mutually overlapping crop plants and high weed pressure. Moreover, single plants can be fragmented by straw or weeds.

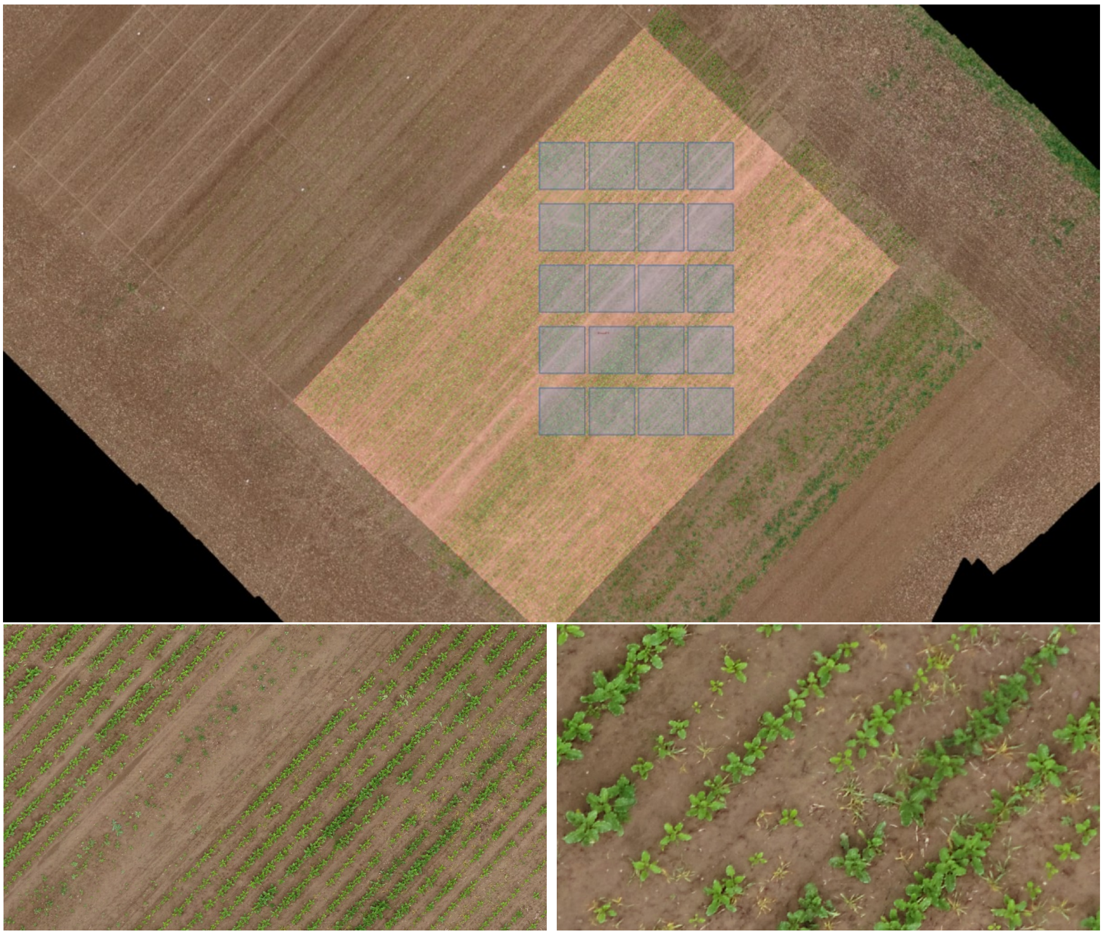


Figure 3.16: Overview of the BONN-UAV-17-5MM dataset. Top: North-oriented orthomosaic of the field. The red bounding box illustrates the area of interest. The blue boxes refer to patches for which we provide pixel-wise labeling into the classes crop, weed, and soil. Bottom left: medium zoomed view showing sugar beet plants and weeds. Bottom right: strongly zoomed view revealing sugar beet plants that have different size and color as well as weeds growing between and in the crop.

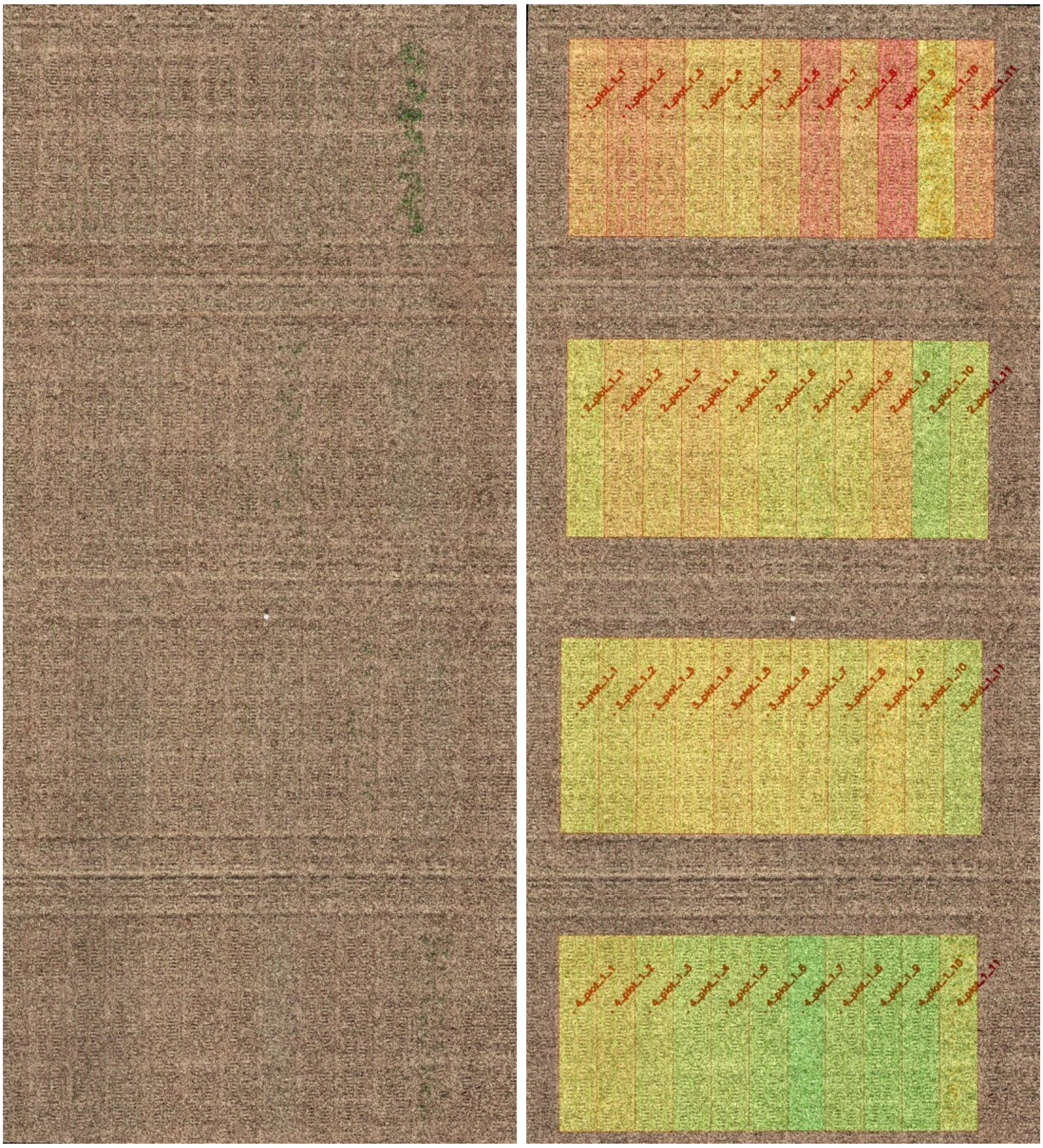


Figure 3.17: Overview of the GOETT-UAV-19 dataset. Left: orthomosaic of the trial field consisting of 44 micro-plots containing sugar beet and weeds. Right: illustration of micro-plots. The colors refer to the average size of the sugar beets. Green refers to small and red refers to big plants regarding the average growth stage.

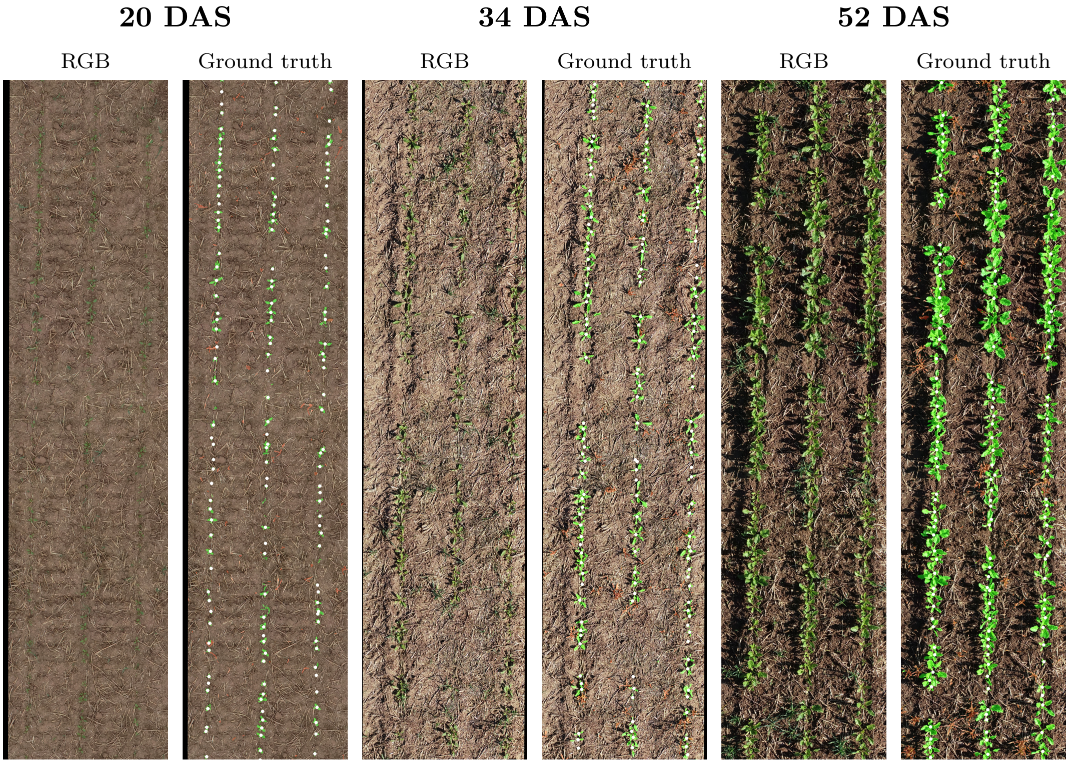


Figure 3.18: Exemplary micro-plots with corresponding ground truth for crop-weed classification and stem detection. The classifier has to deal with mutually overlapping crop plants and high weed pressure. Moreover, single plants can be fragmented through straw or weeds.

GOETT-UAV-19: In autumn 2019, we acquired this dataset in collaboration with the Institut für Zuckerrübenforschung and ARGE NORD who supported the data collection and field management. The field is located near Göttingen in Germany. As for the BONN-UAV-17-5MM dataset, which we described in the previous section, we do not rely on single images but on the resulting orthomosaic. We used the Phantom 4 RTK for data acquisition and flew at an altitude of around 9 m resulting in a ground sampling distance of about 1.5 mm. The plants in this dataset are sown in a specific pattern. The field is divided into 40 micro-plots for which we want to know the exact number of plants inside. Figure 3.17 depicts the field and the plot structure as well as a detailed view of one of the plots.

This dataset consists of three measurements of a field with 40 micro-plots each, i.e., 120 micro-plots in total. Each plot has a size of 4,066 pixels × 985 pixels, and has a spatial extent of around 6 m×1.5 m. We collected the first measurement 20 days after seeding (DAS-20) the plants, shortly after the plants had emerged. Subsequently, two further measurements were carried out ten days after the previous measurement, i.e., DAS-34 and DAS-52. Figure 3.18 depicts an exemplary micro-plot for each measurement day and illustrates the aforementioned challenging conditions for the vision-based plant counting task. Already at the early growth stage, the sugar beets overlap each other along the crop row. Also, individual plants appear as separate components in the image-space because they are covered and separated by straw or larger grown weeds.

We perform ground-truthing by manually counting the plants for all 120 plots in the images. We acquire the actual number of plants per plot for each of the three measurement dates separately. It can happen that over time, new plants appear in a plot due to post-emerge or plants disappear due to death or removal by animals. On average, we counted 203 plants per micro-plot with a standard deviation of around 18 plants considering all micro-plots across the three measurement days. Figure 3.18 illustrates a zoomed view of an example plot per measurement day. For the classifier training, we fully labeled five plots per measurement day. We pixel-wisely labeled the crop plants and weeds as well as the stem locations for the crop plants.

Chapter 4

Plant Classification using Random Forests

The main objective of this thesis is the development of innovative vision-based plant classification systems for agricultural robots allowing the robots to identify the value crop and distinguish it from weeds. Our key developments focus on plant classification systems that enable UGVs for online in-field interventions and enable UAVs to be used for accurate plant monitoring applications.

In this chapter, we introduce our first series of plant classification systems enabling UGVs and UAVs to perceive crop plants and weeds in agricultural field environments. All classification systems we present in this chapter use random forests [15] as their core machine-learning model and are based on handcrafted features. The classification systems operate with RGB images, i.e., $\mathbf{I}_{RGB}$, as well as with 4-channel images, which consist of $\mathbf{I}_{RGB}$ plus an additional near-infrared measurement per pixel, i.e., $\mathbf{I}_{NIR}$. The goal is to map the image input into a label map that encodes a class label for every pixel.

We illustrate the key processing steps of the random forest-based approaches in Figure 4.1. First, we preprocess the $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images and separate the vegetation from the remaining parts of the image, i.e., the image background representing mostly soil.

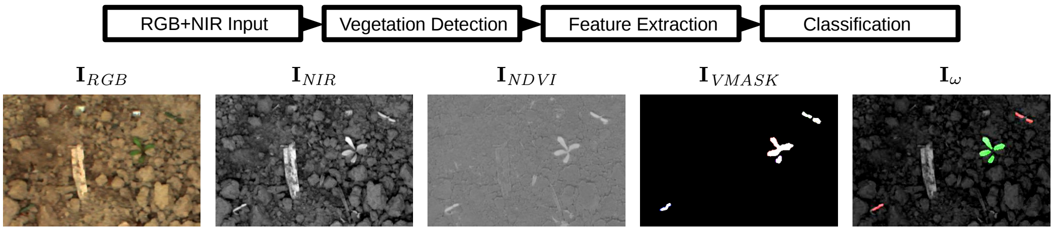


Figure 4.1: Pipeline for the random forest-based classification. From left to right: $\mathbf{I}_{RGB}$ and $\mathbf{I}_{NIR}$ input images, computed $\mathbf{I}_{NDVI}$ image according to Equation (4.2), computed vegetation mask according to Equation (4.1), and the classification output containing sugar beets (green) and weeds (red).

Then, we compute a series of handcrafted features, describing solely the image regions that correspond to vegetation. Third, we use the extracted features to train and deploy a random forest to perform the plant classification. We consider two variants of features for the classification. The first variant computes local features for keypoints and classifies the area around each keypoint. The second variant is an object- or segment-based classification that makes the decision for all pixels in a vegetation segment.

A closer look at this pipeline reveals that there are two classification problems connected in series. The first problem is a binary classification of vegetation and soil. The second problem is a classification of identified vegetation from the first step into crop plants and weeds. This two-step procedure has advantages and disadvantages. The advantage is that in most cases, the vegetation can be separated with comparatively simple threshold operations based on the color information exploiting vegetation indexes.

This simplifies the subsequent plant classification problem, and at the same time, this drastically reduces the number of pixels to be analyzed drastically, since most of the pixel positions in the image data belong to the soil, compare Section 3.2. The disadvantage of this two-step approach is that the random forest-based plant classification step can no longer correct errors in vegetation classification.

4.1 Different Random Forest-Based Classification Systems

Throughout this chapter, we propose four different approaches for the random forest-based plant classification. We first introduce a uniform notation concerning the approaches presented below. We refer to a random forest-based approach with RF-*, where RF stands for the random forest. The postfix is a placeholder for abbreviation referring to the respective variant. All approaches follow the main pipeline shown in Figure 4.1 and use handcrafted features, mainly visual information. We will also introduce modifications of these approaches that exploit additional geometric features about the local arrangement of plants in the field to distinguish plants and weeds.

We start with two approaches that exploit two different ways to address the feature extraction for the classification problem. The first approach extracts local features for keypoints and classifies the area around each keypoint. We refer to this approach as RF-KP. The second variant is an object- or segment-based classification that chooses for all pixels in a vegetation segment. We refer to this approach as RF-OBJ.

Please note that we do not claim a contribution to the RF-KP approach in this thesis, as we developed this approach within the context of the master thesis by Lottes [81]. However, we propose other approaches that build upon RF-KP. Thus, we also explain RF-KP.

Both methods RF-KP and RF-OBJ have their advantages and disadvantages. However, We find a way to link these two approaches exploiting both schemes for feature extraction. We propose the RF-CAS approach, which is our first contribution in this thesis. RF-CAS combines RF-OBJ and RF-KP in a cascade and exploits their respective advantage and even compensates for their respective disadvantages. In Section 4.3, we present our purely visual plant classification system RF-KP, RF-OBJ, and RF-CAS.

The visual features used in these approaches rely on the image intensities encoding color and shape information of the plants. Besides, we also exploit additional geometric information about the local arrangement of the plants in the field. Specifically, we aim at bridging the performance gap of purely visual crop-weed classifiers regarding their generalization capabilities to new and changing field conditions, e.g., in situations when the visual appearance of the plants, weeds, and soil in the field has changed between the training of the classifier and its deployment, see also Section 1.2.1. Therefore, we introduce a probabilistic model representing the spatial arrangement of the crop plants and weeds in the field using coordinate differences between plants. Then, we employ a Bayesian approach to perform the crop-weed classification solely based on geometric features. This leads us to our next proposed approach that combines the visual random forest with the geometric Bayesian classifier in a semi-supervised way and is called RF-GC. This approach reflects a further key contribution in this thesis and is presented in Section 4.4.

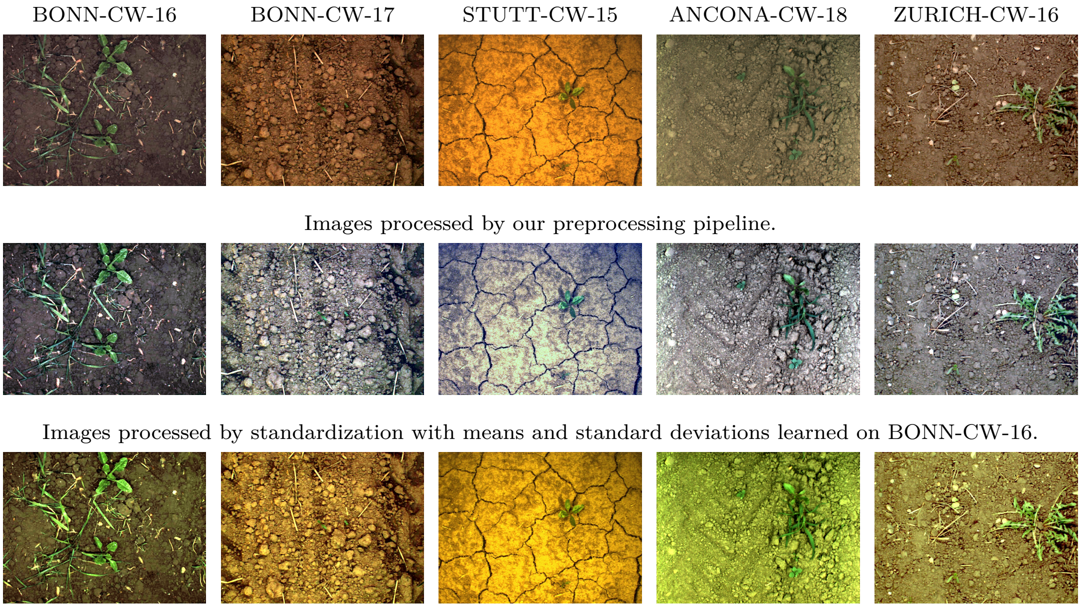
Finally, we adopt our RF-CAS approach to be used with UAV data, which is our next relevant contribution. We propose our next contribution, i.e., RF-UAV, which is based on our RF-CAS approach but considers further geometric features exploiting the field geometry in terms of crop row information and spatial relationships among multiple individual plants. We describe our UAV-based approach in Section 4.5.

Within our experimental Chapter 6, we evaluate the different approaches and their respective properties as well as our key design decisions of the classification models. We presented the RF-KP approach in [86], RF-OBJ and RF-CAS in [87], the RF-GC approach in [89], and the RF-UAV approach in [88]. We implement all proposed variants of the random forest-based plant classification systems as modules for the Robot Operating System ROS [121] and evaluated them on different real field robots, see Chapter 3. We develop computationally demanding tasks such as image preprocessing that we explain in Section 4.2 and the extraction of handcrafted features on a graphics processing unit (GPU) using the CUDA [24] library. Hence, we achieve a sufficient runtime for the processing of the classification results that are required for online in-field operations such as selective spraying or mechanical weed control.

4.2 Preprocessing of the Input

The preprocessing of the image data is the first step of the plant classification pipeline. The procedure for preprocessing of $\mathbf{I}_{RGB}$ and $\mathbf{I}_{NIR}$ images described in this section is the same for *all* approaches presented in this chapter as well as Chapter 5.

In order for classifiers to deliver high performance on different data, it is recommended to preprocess the input data. In preprocessing steps, we apply transformations to the data to reduce its complexity and to standardize it to some degree, thereby



Images processed by our preprocessing pipeline.

Images processed by standardization with means and standard deviations learned on BONN-CW-16.

Figure 4.2: Top row: $\mathbf{I}_{RGB}$ images from different datasets containing different crop growth stages, weed types, soil types, and acquired under different illumination conditions. Middle row: Preprocessed $\mathbf{I}_{RGB}$ images by our approach. Bottom row: The standardization step is performed with the channel-wise means and standard deviations that are learned on the entire BONN-CW-16 dataset. We qualitatively notice that the data distributions across different datasets are more similar by using our proposed preprocessing.

increasing the chance that the machine-learning algorithm can provide better performance than without preprocessing the data. Technically speaking, preprocessing can help to improve the generalization capabilities of a classification system by aligning the training and test data distribution.

In this thesis, we are dealing with images, which are recorded by the same camera system, but under different lighting conditions and in different fields. These factors may lead to the fact that the colors in the images can be distributed differently across different datasets. To illustrate this effect, Figure 4.2 shows some example $\mathbf{I}_{RGB}$ images from different UGV data sets. The primary objective is to train classification models that perform well under similar, but also under changing field conditions. Thus, the goal of our preprocessing pipeline is to minimize the diversity in color across different datasets.

We perform the preprocessing independently for each image and separately on all channels, i.e., red, green, blue, and near-infrared. For each channel, we first remove noise by performing a blurring operation using a $[5 \times 5]$ Gaussian kernel given by the standard normal distribution, i.e., $\mu = 0$ and $\sigma^2 = 1$. Second, we standardize each image channel by its mean and standard deviation, respectively. Third, we perform a contrast stretch of the intensities to the range $[-0.5, 0.5]$, which implies a zero-centering of the data. Figure 4.2 illustrates the effect of our preprocessing for exemplary images captured with different sensor setups.

We preprocess each image independently and standardize its respective channel

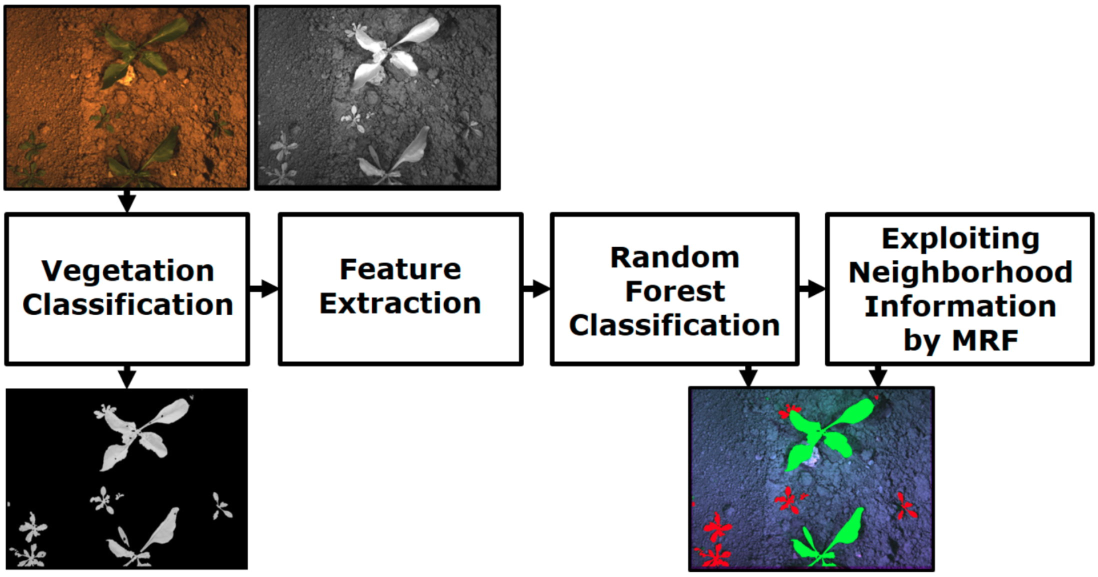

Vegetation Classification
Feature Extraction
Random Forest Classification
Exploiting Neighborhood Information by MRF

Figure 4.3: Classification pipeline for the RF-KP and RF-OBJ approach. First, we preprocess the input and classify the vegetation. Then, we extract keypoint-based or object-based features describing the vegetation and classify the vegetative image regions using random forests. For the keypoint-based approach RF-KP, we further utilize using a Markov random field to spatially smooth the predicted labels.

means and standard deviations. Thus, there is no learning involved in our method. Figure 4.2 qualitatively illustrates the results achieved with our preprocessing and compares them to the results achieved with preprocessing using the channel-wise means and standard deviations we learned on the entire BONN-CW-16 data set. Looking at the images reveals that our method achieves a better alignment of intensities across the different data sets. In the case of the other method, the effect of matching to BONN-CW-16 is barely visible. Also, quantitatively, our results in our experimental evaluation in Section 6.6.3 show that our method for preprocessing achieves a substantial improvement for the generalization capabilities to new or changing conditions.

4.3 Vision-Based Plant Classification using Random Forests

In this section, we discuss the purely visual plant classification systems, i.e., RF-KP, RF-OBJ, and RF-CAS. Figure 4.3 depicts the principal processing pipeline for the RF-KP and RF-OBJ approach. The main goal of the visual plant classification system is to provide a pixel-wise classification of the scene. The input to our classifiers is $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images. The output of the classification system is a label map considering the classes $\omega^{cws} \in \{\omega_{\mathsf{c}}, \omega_{\mathsf{w}}, \omega_s\}$ for crop, weed, and soil.

Our overall pipeline works in five steps: First, we preprocess the images according to Section 4.2. Second, we identify the vegetation by performing binary pixel-wise classification of the input. This procedure leads to a vegetation mask $\mathbf{I}_{VMASK}$, see

Figure 4.4 (bottom right) for an example. This step is highly effective, as it allows us to compute the features on the subsequent processing steps only for the regions that correspond to vegetation. Third, we compute a set of features for the image regions that correspond to vegetation. Here, we follow two different approaches: (i) a keypoint-based approach and (ii) an object-based approach. The keypoint-based approach computes local features on a dense grid of keypoints and performs the classification for each keypoint. This procedure is computationally demanding but allows us to handle the situation in which two plants are close to each other or overlap. The object-based approach performs one classification per segment (object) and thus is substantially faster to compute but cannot handle overlapping plants well. Forth, we classify the keypoints or objects employing random forests, which yields a probability distribution representing the fact that the area under consideration corresponds to our crop or to weed. We call the approach classifying the keypoints RF-KP and the approach classifying segmented objects RF-OBJ. Finally, we improve the classification for RF-KP through exploiting neighborhood information using a Markov random field. Thereby, we spatially smooth the random forest's labeling and reduce the number of wrongly classified keypoint. In the Subsections 4.3.1-4.3.5, we provide a detailed description of these steps. Finally, we discuss how to combine the keypoint-based and object-based classification.

4.3.1 Vegetation Classification

The goal of vegetation detection is to eliminate the irrelevant background from the image so that the subsequent classification task operates on regions that correspond to vegetation. The $\mathbf{I}_{NIR}$ information is especially useful for separating the vegetation from the soil and other backgrounds due to the high reflectivity of chlorophyll and thus (healthy) plants in the $\mathbf{I}_{NIR}$ spectrum [127]. We compute a vegetation mask.

$$\mathbf{I}_{VMASK}(i,j) = \begin{cases} 1, & \text{if } \mathbf{I}(i,j) \in \text{vegetation} \\ 0, & \text{otherwise} \end{cases}, \tag{4.1}$$

with the pixel location (i, j). To separate the vegetation, we exploit specific reflectance of healthy vegetation using the normalized difference vegetation index ($\mathbf{I}_{NDVI}$) according to [127] using the $\mathbf{I}_{NIR}$ channel $\mathbf{I}_{NIR}$ and the red channel $\mathbf{I}_{R}$ on a per-pixel basis:

$$\mathbf{I}_{NDVI}(i,j) = \frac{\mathbf{I}_{NIR}(i,j) + \mathbf{I}_{R}(i,j)}{\mathbf{I}_{NIR}(i,j) - \mathbf{I}_{R}(i,j)}. \tag{4.2}$$

Figure 4.4 (top right) shows an example of a $\mathbf{I}_{NDVI}$ image $\mathbf{I}_{NDVI}$ for sugar beet plants and weeds. In the field, the reflectivity of chlorophyll typically leads to a bimodal intensity distribution in $\mathbf{I}_{NDVI}$ for healthy vegetation and allows us to perform a threshold-based classification on the $\mathbf{I}_{NDVI}$ information for every pixel. Figure 4.4 (bottom left) depicts the $\mathbf{I}_{NDVI}$ intensity distribution for an example image. Next, we perform a the threshold-based vegetation classification, i.e.,

$$\mathbf{I}^{*}_{VMASK}(i,j) = \begin{cases} 1, & \text{if } \mathbf{I}_{NDVI}(i,j) \geq t \\ 0, & \text{otherwise} \end{cases}, \tag{4.3}$$

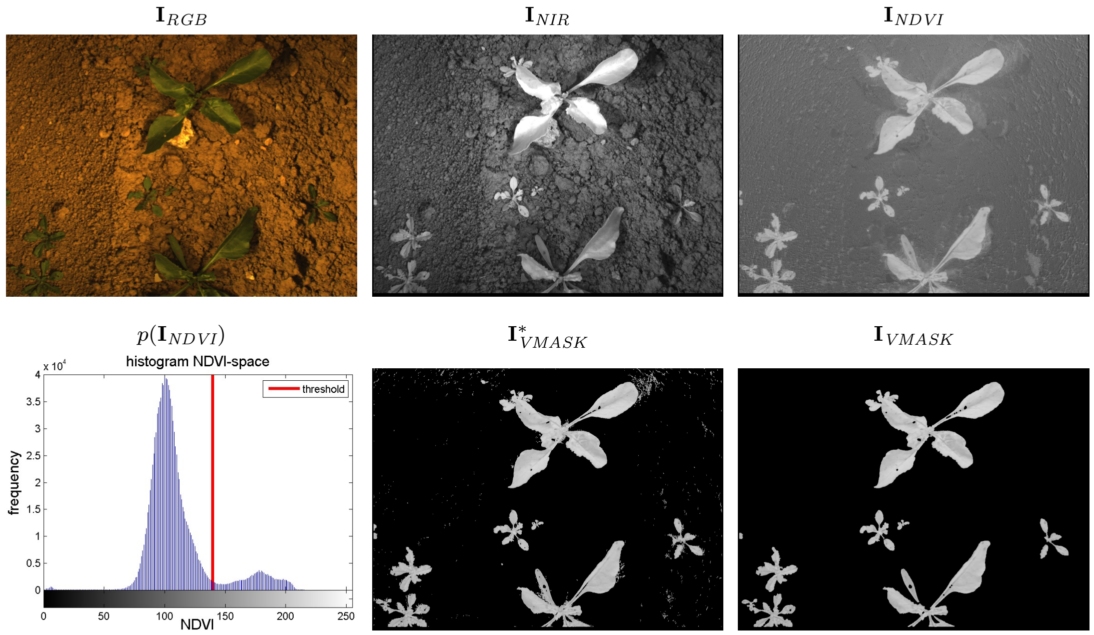


Figure 4.4: From left to right (Top): Raw input $\mathbf{I}_{RGB}$ image, $\mathbf{I}_{NIR}$ image, and processed $\mathbf{I}_{NDVI}$ image. From left to right (Bottom): Histogram of $\mathbf{I}_{NDVI}$ values and selected threshold t (red) for classifying the vegetation according to Equation (4.3), masked $\mathbf{I}_{NDVI}$ after threshold operation containing errors, final vegetation mask after optimization.

where 1 refers to vegetation and 0 refers to the background (mostly soil). Here, we use the $\mathbf{I}_{NDVI}$ as vegetation index, but different representations are possible.

A threshold-based classification based on the $\mathbf{I}_{NDVI}$ may lead to small residual errors. Examples for such small errors are visible on the top right area of the middle image in Figure 4.4. These effects are often caused by lens errors, especially chromatic aberration, resulting in slightly different mappings of the red and the near-infrared light from the workspace to pixels on the chip. Most of the residual errors can be eliminated through basic image processing techniques such as (i) requiring a minimum brightness in $\mathbf{I}_{NIR}$, (ii) using morphological opening and closing to fill gaps and to remove noise at contours, and (iii) removing regions conatining a few pixels only. Figure 4.4 (bottom right) depicts the final application of the vegetation mask $\mathbf{I}_{VMASK}$ on the $\mathbf{I}_{NDVI}$ image.

The vegetation classification is explicitly tested in our experimental evaluation in Section 6.4, since it plays an important role in the whole system for all approaches in this chapter. We first investigate which vegetation index, e.g., the $\mathbf{I}_{NDVI}$, is particularly suitable for the separation of vegetation and soil. We anticipate the investigations at this point. We use the $\mathbf{I}_{NDVI}$ index if we have access to $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images and the Excess Green Index ($\mathbf{I}_{ExG}$) given by

$$\mathbf{I}_{ExG} = 2\,\mathbf{I}_G - \mathbf{I}_R - \mathbf{I}_B, \tag{4.4}$$

if we only have access to $\mathbf{I}_{RGB}$ images, as is usually the case with UAVs. This decision is made based on empirical evaluation. The NDVI and ExG show superior performance for the vegetation classification under similar and changing field conditions.

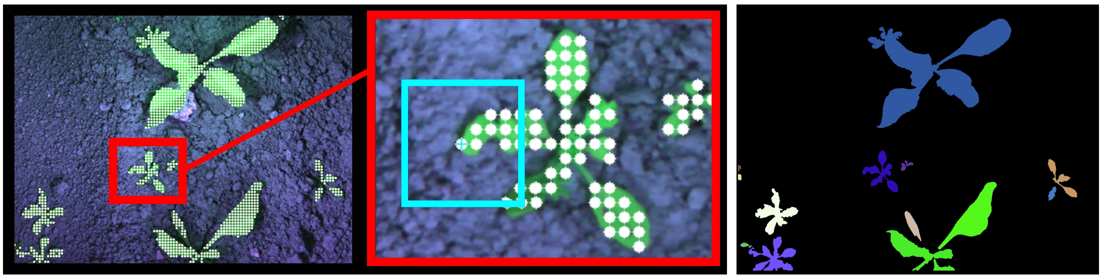


Figure 4.5: Left: Keypoints $\mathcal{K}$ (white) for classification at a 3 mm distance on the object. Middle: Zoomed view depicting the neighborhood $\mathcal{P}_{\mathcal{K}}$ of a keypoint $\mathcal{K}$ (blue) representing the region that is considered for the local feature computation for that keypoint. Right: Segmented vegetation. The segments define individual objects that are used for the object-based classification. Here, we compute the features for each segment globally and perform a single classification per segment and not per keypoint.

4.3.2 Keypoint-Based vs. Object-Based Classification

We propose two different ways to address the feature extraction for our classification problem. First, we can compute features for each keypoint and perform the classification for each keypoint individually. Second, we can perform an object-based approach. Here, we define objects as segments of the vegetation pixels through connected components and perform only one classification per object. See Figure 4.5 for an example.

Keypoint-Based approach Please note that we do not claim a contribution to the RF-KP approach in this thesis, as we developed this approach with the context of the master thesis by Lottes [81]. The keypoint-based approach computes features and performs the classification for each keypoint individually. It has the advantage that it can deal with plants that overlap but at the cost of being computationally expensive. A typical setup in our experimental evaluation in Chapter 6 is that keypoints are spaced 10 pixels by 10 pixels apart. To extract information about the class label of each keypoint, we use a fixed-sized neighborhood to compute the features. In our current implementation, the neighborhood $\mathcal{P}_{\mathcal{K}}$ of a keypoint $\mathcal{K}$ has a size of 40 pixels by 40 pixels. Figure 4.5 (left, middle) illustrates the arrangement of keypoints on an image including the neighborhood for feature extraction. Details on the features are given in Section 4.3.3. In the remainder of this thesis, we refer to this approach as RF-KP.

Object-Based Approach Alternatively, we perform an object-based approach. Each object $\mathcal{O}$ is given through a connected component of the classified vegetation pixels in the computed vegetation mask $\mathbf{I}_{VMASK}$. Thus, we can perform one classification per object. Figure 4.5 (right) depicts examples of found objects. This approach has the advantage that it is substantially faster than the keypoint-based approach, as fewer samples describing the whole vegetation within an image need to be analyzed but suffers from situations in which weed and value crop overlap. In the remainder of this thesis, we refer to this approach as RF-OBJ.

4.3.3 Feature Extraction

Both approaches share the concept of partitioning vegetation into parts that are classified separately and thus lead to the same feature extraction procedure, except that the areas in which the features are computed differ. We extract a set of features $\mathcal{V}$ for each approach, either for $\mathcal{P}_{\mathcal{K}}$ or for the whole object $\mathcal{O}$. We categorize the features into three groups, which are explained in the remainder of this section. Note that for some features, we use the $\mathbf{I}_{NDVI}$ distribution as the basis for the features. Analogous to the vegetation detection step, we use the $\mathbf{I}_{ExG}$ distribution instead of the $\mathbf{I}_{NDVI}$ if only the $\mathbf{I}_{RGB}$ information is available.

Statistical features Our set $\mathcal{V}_{St}$ of statistical features includes the following parameters for describing the distribution of the inputs. Here, we use: min, max, range, mean, standard deviation, median, skewness, kurtosis, and entropy. These statistical means are computed on different input sources, which are given by the different input channels of our image data as well as gradient representations, and texture information. The set of input sources is defined by

$$\mathcal{S} := \{\mathbf{I}_x, \nabla\mathbf{I}_x, \Delta\mathbf{I}_x, LBP(\mathbf{I}_x), LBP(\nabla\mathbf{I}_x), LBP(\Delta\mathbf{I}_x)\}, \tag{4.5}$$

where $\mathbf{I}_x$ are different channels of the image data, $\nabla\mathbf{I}_x$ are its gradients, $\Delta\mathbf{I}_x$ are the Laplacians, and *LBP* are the locally binary pattern representations encoding texture information. All these quantities are defined in more detail in the remainder of this section. First, we convert our four raw input channels $\mathbf{I}_R$, $\mathbf{I}_G$, $\mathbf{I}_B$, and $\mathbf{I}_{NIR}$ into the following six channels

$$\mathbf{I}_x \quad \text{with} \quad x = \{\mathbf{I}_{NDVI}, \mathbf{I}_G, \mathbf{I}_B, \mathbf{I}_H, \mathbf{I}_S, \mathbf{I}_L\}. \tag{4.6}$$

Here, $\mathbf{I}_{NDVI}$ is the normalized difference vegetation index as defined in Equation (4.2), while G and B are the green channels of our images. The subscripts H, S, and L refer to the Hue-Saturation-Lightness (HSL) representation, which is a variant of the HSV color space, which is frequently used for plant and leaf classification, e.g., [29, 70, 134]. The HSL(I_1, I_2, I_3) color space represents the three input channels I_1, I_2, I_3 as cylindric coordinates and separates intensity from color information. The dimension L called lightness is defined by

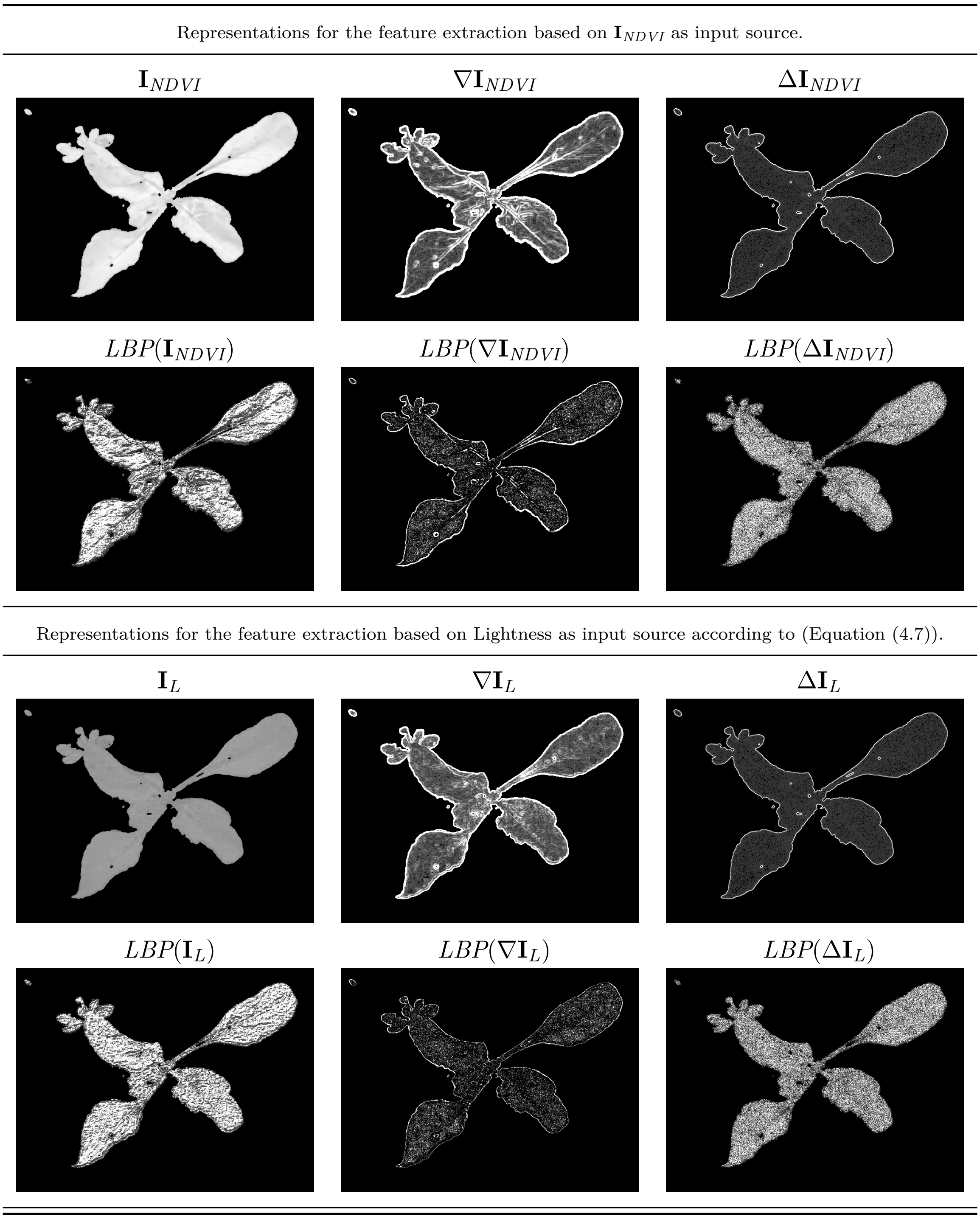
$$L = \frac{\max(I_1, I_2, I_3) - \min(I_1, I_2, I_3)}{2}. \tag{4.7}$$

We use the HSL space instead of the HSV space (lightness $\mathbf{I}_L$ instead of value) because it is related to the average range of the input intensities and, therefore, more robust concerning biased intensities. Throughout this work, we define the HSL channels as

$$\mathbf{I}_{HSL} = \text{HSL}(\mathbf{I}_{NDVI}, \mathbf{I}_G, \mathbf{I}_B). \tag{4.8}$$

For each input source $\mathbf{I}_x$, we also consider its gradients $\nabla\mathbf{I}_x$ and the Laplacian (second-order gradients) Δx. The magnitudes of $\nabla\mathbf{I}_x$ and $\Delta\mathbf{I}_x$ provide information about structure and homogeneous regions and are computed by:

$$\nabla\mathbf{I}_x = \left|\frac{\partial\mathbf{I}_x}{\partial i}\right| + \left|\frac{\partial\mathbf{I}_x}{\partial j}\right| \tag{4.9}$$



Representations for the feature extraction based on $\mathbf{I}_{NDVI}$ as input source.

$\mathbf{I}_{NDVI}$ $\nabla\mathbf{I}_{NDVI}$ $\Delta\mathbf{I}_{NDVI}$

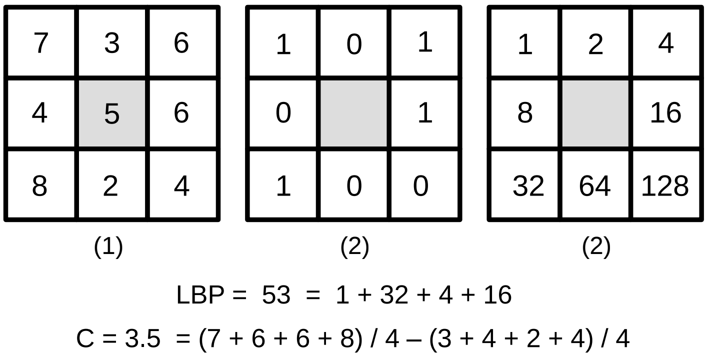
$LBP(\mathbf{I}_{NDVI})$ $LBP(\nabla\mathbf{I}_{NDVI})$ $LBP(\Delta\mathbf{I}_{NDVI})$

Representations for the feature extraction based on Lightness as input source according to (Equation (4.7)).

$\mathbf{I}_{L}$ $\nabla\mathbf{I}_{L}$ $\Delta\mathbf{I}_{L}$

$LBP(\mathbf{I}_{L})$ $LBP(\nabla\mathbf{I}_{L})$ $LBP(\Delta\mathbf{I}_{L})$

Figure 4.6: Different input sources $\mathcal{S}$ according to Equation (4.5) for the statistical feature extraction. For the sake of brevity, we only visualize the representations for the $\mathbf{I}_{NDVI}$ and L input source.



7	3	6
4	5	6
8	2	4

(1)

1	0	1
0		1
1	0	0

(2)

1	2	4
8		16
32	64	128

(2)

LBP = 53 = 1 + 32 + 4 + 16

C = 3.5 = (7 + 6 + 6 + 8) / 4 – (3 + 4 + 2 + 4) / 4

Figure 4.7: Example for the computation of a LBP number and the corresponding contrast measure C for a pixel given its 8-connected neighborhood. (1) input, (2) threshold operation by value of center pixel, and (3) binomial weights according to [105].

and

$$\Delta \mathbf{I}_x = \left| \frac{\partial^2 \mathbf{I}_x}{\partial i^2} + \frac{\partial^2 \mathbf{I}_x}{\partial j^2} \right| \tag{4.10}$$

Finally, we take into account distributions of texture information and contrast. These distributions are based on local binary patterns, according to [105]. The LBP operator performs thresholding operations within a 8-connected neighborhood based on the value of its center pixel and converts this pattern as a binary number. Figure 4.7 illustrates the computation of an LBP number and the associated contrast measure C for a pixel.

The computation of our nine statistical features $\mathcal{V}_{St}$ on all input sources $\mathcal{S}$ leads to 324 statistical features per keypoint or object. We also use a combination of features, which turned out to be useful for the crop-weeds classification problem. Specifically, a ratio between the entropy of the first-order gradient of the $\mathbf{I}_{NDVI}$ image and its Laplacian:

$$\mathcal{V}_{17} = \frac{\mathcal{V}_9(\nabla \mathbf{I}_{NDVI})}{\mathcal{V}_9(\Delta \mathbf{I}_{NDVI})} \tag{4.11}$$

A summary of the statistical features and input sources is given in Table 4.1.

Shape Features The next set of features describes different aspects of the plant's shape. As for the statistical features, we compute shape features for a local neighborhood $\mathcal{P}_{\mathcal{K}}$ of a keypoint or for an object $\mathcal{O}$. The shape features $\mathcal{V}_{Sh}$ only need to be computed on the vegetation mask $\mathbf{I}_{VMASK}$, which is a binary image. We consider the following features describing contours, relations to geometric primitives, and geometric ratios:

- Rectangularity of the contour using its major a and minor b axes of the minimum enclosing oriented ellipse.

$$\mathcal{V}_{13} = \frac{\text{area}}{a\,b}, \tag{4.12}$$

  where area refers to the area covered by vegetation within the patch or the size of the object.

- Aspect ratio of the major a and minor b axes of the minimum enclosing oriented ellipse of the contour:

$$\mathcal{V}_{14} = \frac{a}{b} \tag{4.13}$$

- Area change under smoothing, which describes how the area of vegetation changes due to a smoothing with different-sized Gaussian kernels G and is given by the ratio of the areas:

$$\mathcal{V}_{15} = \frac{\text{area}(G_{\mathbf{I}_{VMASK}}(\sigma))}{\text{area}(G_{\mathbf{I}_{VMASK}}(2\sigma))} \tag{4.14}$$

- Form factor F, which provides a measure of the shape of an object:

$$\mathcal{V}_{16} = \frac{4\,\pi\,\text{area}}{\text{perimeter}^2}, \tag{4.15}$$

where perimeter is the perimeter of the area that is covered by the vegetation. We additionally exploit the convexity, compactness, and solidity feature as described in [50]. Table 4.1 gives a summary of all features used in our classification system and described based on which inputs they are computed.

4.3.4 Random Forest Classification

For the classification, we apply a random forest [15] because it provides comparably robust classification results. As an ensemble method, random forests reduce the risk of overfitting to some degree and can implicitly estimate confidences for the class labels. Regarding Equation (2.7), a pseudo probability $p(\boldsymbol{\omega} \mid \Phi(\boldsymbol{V}, \Theta))$ for a predicted class label ω can be estimated by considering the outputs of the individual decision-tree classifiers within a set of decision trees called the ”forest”. In addition to that, random forests are capable of solving multi-class problems. Mainly, we are interested in distinguishing two classes, i.e., the “crop”, referred to as ω_c, and the “weed” class ω_w. For the object-based approach, however, we introduce an additional class, “mixed” (ω_m), for segmented objects that contain both weeds and sugar beets, as the plants overlap. This is only relevant for the object-based approach and not for the keypoint-based one. In our implementation, we use all cores of our CPU by running the individual trees of the forest in different threads. A detailed description of the mathematical model and the training procedure of random forests is given in Section 2.1.2. We refer to approaches using the random forest for the classification task with the RF prefix.

The three steps (i) vegetation detection, (ii) feature extraction, and (iii) random forest classification, lead to labeled images such as the ones shown in Figure 4.8 and provide results that are already sufficient for weed control applications, but still, contain errors indicated by blue arrows. The keypoint-based approach falsely classifies some data points, whereas the object-based approach is not able to accurately classify the entire vegetation in the example image by design. The sugar beet plant (green) and

Table 4.1: Features used by our classification system

Nr.	Feature set
	Statistical features $\mathcal{V}_{St}(S)$
$\mathcal{V}_1$	min
$\mathcal{V}_2$	max
$\mathcal{V}_3$	range
$\mathcal{V}_4$	mean
$\mathcal{V}_5$	standard deviation
$\mathcal{V}_6$	median
$\mathcal{V}_7$	skewness
$\mathcal{V}_8$	kurtosis
$\mathcal{V}_9$	entropy
	All statistical features $\mathcal{V}_{St}(S)$ are computed from input sources in $S = \{\mathbf{I}_x, \nabla\mathbf{I}_x, \Delta\mathbf{I}_x, LBP(\mathbf{I}_x), LBP(\nabla\mathbf{I}_x), LBP(\Delta\mathbf{I}_x)\}$ with $x = \{\mathbf{I}_{NDVI}, \mathbf{I}_G, \mathbf{I}_B, \mathbf{I}_H, \mathbf{I}_S, \mathbf{I}_L\}$ this leads to 324 statical features, i.e.: 9 features on 6 channels on 6 input sources
	Shape features $\mathcal{V}_{Sh}$ computed on binary image $\mathbf{I}_{VMASK}$
$\mathcal{V}_{10}$	Convexity
$\mathcal{V}_{11}$	Compactness
$\mathcal{V}_{12}$	Solidity
$\mathcal{V}_{13}$	Rectangularity
$\mathcal{V}_{14}$	Aspect ratio of the minimum enclosing ellipse
$\mathcal{V}_{15}$	Area change under smoothing
$\mathcal{V}_{16}$	Form factor
	Other features
$\mathcal{V}_{17}$	$\mathcal{V}_9(\nabla\mathbf{I}_{NDVI})/\mathcal{V}_9(\Delta\mathbf{I}_{NDVI})$

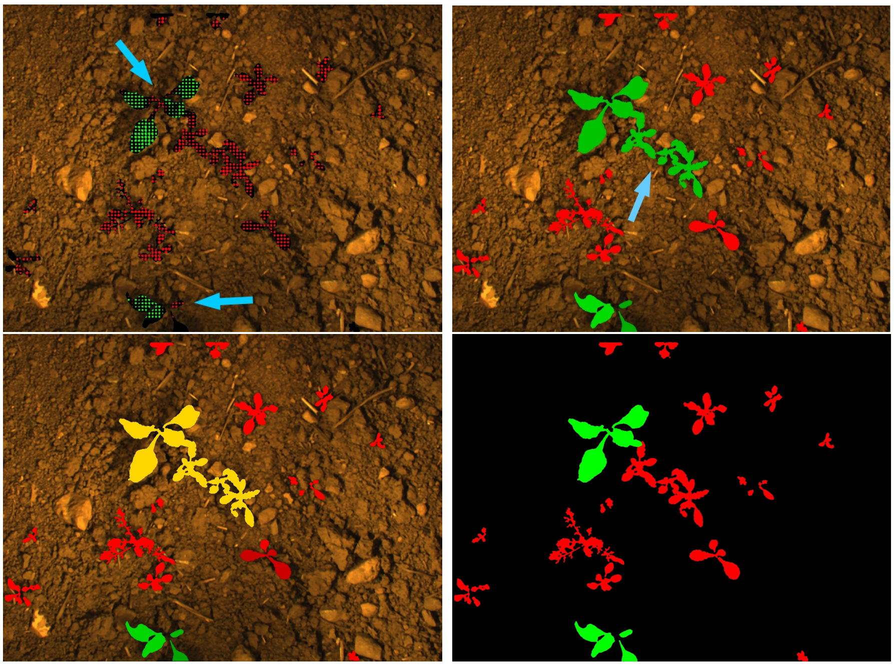


Figure 4.8: From left to right (top): Classification results considering ω_c (green) and ω_w (red) on top of $\mathbf{I}_{RGB}$ obtained by the RF-KP (keypoint-based) approach and by the RF-OBJ (object-based) approach. Classification errors are illustrated by blue arrows. From left to right (bottom): RF-OBJ approach considering ω_c, ω_w, and ω_m (orange) and ground truth map.

weeds (red) are considered to belong to one object, as an under-segmented connected component represents the vegetation. To explicitly deal with those kinds of objects, we introduce the mixed class (orange). A further limitation of the approach is that actual vegetation, which is not detected during the vegetation detection step, is considered to be soil and not further analyzed by the random forest classification.

4.3.5 MRF Smoothing for Keypoint-Based Classification

This section is only relevant for the RF-KP approach and does not apply to the object-based one. The keypoint-based classification system described so far computes each label assignment independently of the other nearby labels. In order to improve the classification results and to exploit the topological relationships between keypoints, we apply a Markov random field (MRF). We compute a global classification based on the individually computed class labels $\omega(\mathcal{K})$ of the keypoints by considering their spatial distribution and class confidences $p(\boldsymbol{\omega}(\mathcal{K}) \mid \Phi(\boldsymbol{V}, \Theta))$. We achieve this by minimizing the energy function

$$E(\omega(\mathcal{K})) = \sum_{\mathcal{K}} \left(A(p(\boldsymbol{\omega}(\mathcal{K}) \mid \Phi(\boldsymbol{V}, \Theta))) + \sum_{\mathcal{K}' \in \mathcal{N}_4(\mathcal{K})} B(\omega(\mathcal{K}'), \omega(\mathcal{K})) \right) \tag{4.16}$$

through belief propagation. Here, $E(\omega(\mathcal{K}))$ describes the quality of labeling under the key assumption that neighboring labels vary slightly, but can also change erratically

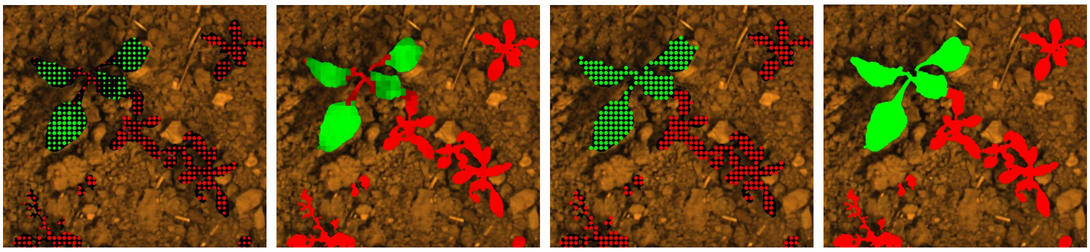


Figure 4.9: Left to right: keypoint-based random forest classification, interpolation of classification results to full image resolution leading to label mask $\mathbf{I}_\omega$, keypoints after spatial smoothing with MRF, and final semantic segmentation result after interpolation of the MRF results. The MRF smoothing eliminates the few wrongly classified keypoints at the plant stem region and outliers.

at class borders. Therefore, two energy terms are needed. The first one, A, considers the confidence of a class label, and through this defines the energy which is needed to change the label. The term B describes the energy for smoothing the four-connected neighborhood, i.e., how many neighboring labels agree. We minimize Equation (4.16) using belief propagation [30]. The MRF optimizes the classification results, as it reduces wrong local estimates by exploiting neighborhood information and considers the confidence of the individual keypoint classifications.

In order to obtain the full semantic segmentation, which is a prediction per pixel instead of per keypoint, we perform a straightforward nearest-neighbor interpolation of the predicted class labels concerning the vegetation mask between the keypoints. This leads to a label mask $\mathbf{I}_\omega$ with the same resolution as the input images. Figure 4.9 depicts a typical example of a sugar beet plant, where MRF optimization leads to better performance. One effect of the MRF smoothing is that wrongly classified plant stem regions, as depicted in Figure 4.9 (left), are corrected. We explicitly evaluate the effect of the MRF on the classification performance in an ablation study in Section 6.3.1.2. As the MRF smoothing provides a performance gain in all cases, we use it as a standard postprocessing step within the RF-KP approach.

4.3.6 Combining Keypoint-Based and Object-Based Classification

To achieve both, the fast execution time of the RF-OBJ approach as well as the ability to deal with overlapping plants of the RF-KP approach, we combine both approaches in a cascade. Through a cascaded classification, we initially apply the object-based approach for the whole image. All objects, which are identified as weeds or sugar beet with high certainty, keep their labeling. For objects with uncertain classification results or which are classified as “mixed” objects, i.e., under-segmented objects due to overlap, we apply the keypoint-based approach. As a result of that, the features only need to be computed for a comparatively small number of keypoints, and thus we can maintain an overall fast computation time. In the remainder of this thesis, we refer to the cascaded classification as RF-CAS. This approach can be seen as the main approach for visual

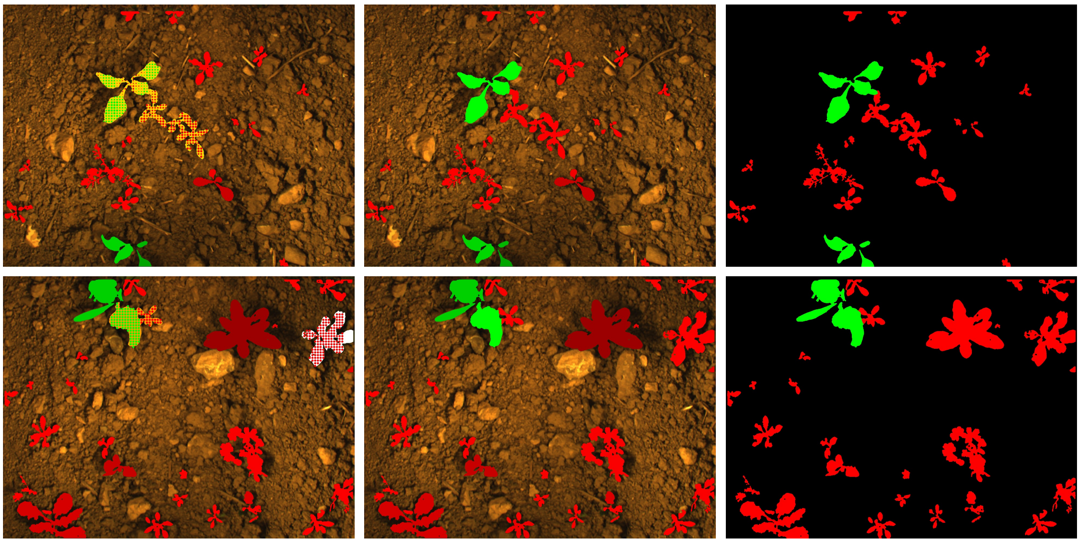


Figure 4.10: Classification results of the RF-KP, RF-OBJ, and RF-CAS approaches. From left to right (top and bottom): (i) classification result based on the cascaded approach, where the detected mixed (orange) as well as uncertain (white) objects are further analyzed by the keypoint-based approach, (ii) final semantic segmentation result, and (iii) corresponding ground truth.

plant classification using random forest with handcrafted features.

In our experimental section (Section 6.3.1.2) we show that both the keypoint-based RF-KP approach as well as the object-based RF-OBJ approach perform well in our datasets and can be combined with the cascaded RF-CAS approach to compensate their drawbacks, respectively,

In more detail, the RF-KP is only executed for objects for which at least one of the two conditions hold. Either the random forests suggests a mixed object

$$\underset{\omega}{\operatorname{argmax}}\, p(\omega \mid \mathcal{V}) = \omega_m, \tag{4.17}$$

or the random forest is too uncertain about is results, i.e.

$$\max\, p(\omega \mid \mathcal{V}) < t^{\mathcal{O}}_{min}, \tag{4.18}$$

where $t^{\mathcal{O}}_{min}$ indicates the minimum probability for the class suggested by the random forest. All those objects are passed to the keypoint-based classifier for a further in-depth investigation.

4.4 Exploiting Plant Arrangement

The geometric signal given by the plant and weed arrangement can be a strong supporter for distinguishing these classes in the field. In row crops, humans typically can perceive the plants and weeds just by analyzing the plant arrangement, even if they are non-experts in the agricultural domain. Figure 4.11 depicts the typical spatial arrangement of crop plants induced by the process of sowing.

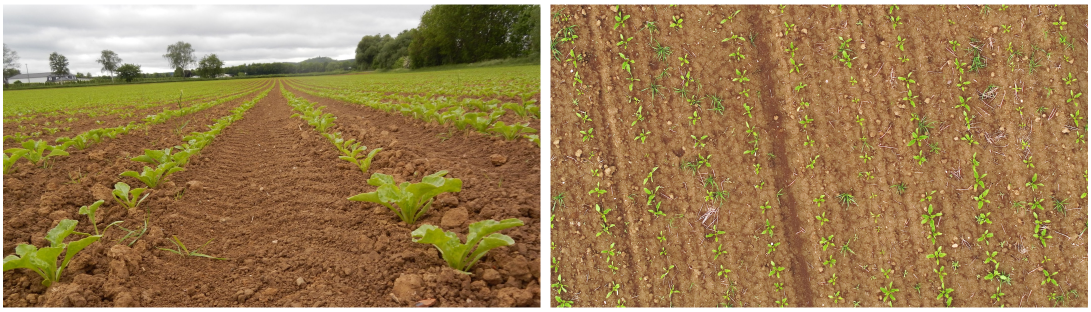


Figure 4.11: Common spatial arrangement of crop plants (sugar beets) in a field. Caused by the process of sowing, the plants are arranged in rows and share a similar spacing between each other along the row. In contrast, weeds appear randomly in the field.

This section is about how we model the plant arrangement and how we gather this information algorithmically from the data to integrate it into our vision-based crop-weed classification system. Specifically, we exploit the pattern of crop plants that are given by its row structure and a similar spacing between the crop plants along the row. In contrast, weeds grow somewhat randomly in the field and can be assumed to follow a uniform spatial distribution.

The key idea is to model the arrangement for the crop plants as well as for weeds as two probability distributions of coordinate differences observed between the plants and to employ a Bayesian approach to obtain a probabilistic output describing the likelihood that a particular vegetation object is a crop or a weed. To incorporate this information, we design an additional and independent classifier to perform the crop-weed classification solely based on the geometric features using a naive Bayesian classification approach. We call this approach the geometric classifier GC. The goal of the geometric classifier is to assign the class labels $\omega = \{c, w\}$ to each detected keypoint $\mathcal{K}$ or object $\mathcal{O}$ based only on spatial information by exploiting the relative arrangement of the plants in the field. Finally, we combine the visual RF-CAS and geometric classifier GC that complement each other through independent predictions and exploit the geometric signal of the spatial crop arrangement to support and retrain the vision-based classifier in a semi-supervised way. We refer to this approach as RF-GC.

4.4.1 Probabilistic Plant Arrangement Model

We define our relative arrangement model through conditional probability distributions

$$p(\mathcal{D} \mid \omega) \quad \text{with} \quad \omega = \{c, w\}, \tag{4.19}$$

of intra-class coordinate differences observed in a coordinate system, for which the x-axis is aligned to the actual crop row direction, where

$$\mathcal{D} = \{\Delta \boldsymbol{x}_1^{row}, \dots, \Delta \boldsymbol{x}_N^{row}\}_{n=1}^{N}, \tag{4.20}$$

is a set of size N consisting of 2D coordinate differences. The intra-class coordinate differences are given by

$$\Delta \boldsymbol{x}^{row} = [|\Delta x^{row}|, |\Delta y^{row}|]^\top \tag{4.21}$$

and measured between the 2D positions of plants. Considering the crop plants, $|\Delta x^{row}|$ is the distance between sugar beet plants along the crop row reflecting the similar spacing between them, and $|\Delta y^{row}|$ is the distance between two sugar beets across the crop row, which tends to take mainly small values around 0, see Figure 4.12. For the computation of $\Delta \boldsymbol{x}^{row}$, we use the positions $\boldsymbol{x}^{row}_{\mathcal{O}}$ of the center of mass for each object $\mathcal{O}$ as the reference point.

4.4.2 Learning the Crop Arrangement from Data

As new data arrives, we perform three steps to learn the crop arrangement model: (i) We represent the detected vegetation in a local map of a fixed size, (ii) estimate the actual crop row considering the already classified crop plants on the local map, and (iii) compute $\mathcal{D}$ according to Equation (4.20) and use it to update the plant arrangement model $p(\mathcal{D} \mid \omega)$. By this, we obtain an up-to-date model representing the probabilities of observing intra-class coordinate differences within a particular local area in the field.

Vegetation Mapping First, we build a map of the segmented objects $\mathcal{O}$ or keypoints $\mathcal{K}$. We use the wheel odometry measurements to determine the motion of the camera and apply a pinhole camera model to project $\mathcal{O}$ to the surface of the field, which we assume to be a plane. For estimating $\mathcal{D}$, we consider only objects $\mathcal{O}$ or keypoints $\mathcal{K}$ that lie within an area of 2 m along the crop row and 0.25 m across the crop row concerning the current position of the camera. The reason for limiting the space along the driving direction is to minimize the effect of drift on the mapping that can be induced through the integration of the wheel odometry measurements over time. The space limitation across the driving direction is based on the camera's footprint on the field surface (see Section 3.1). Figure 4.12 shows a sketch of an obtained map.

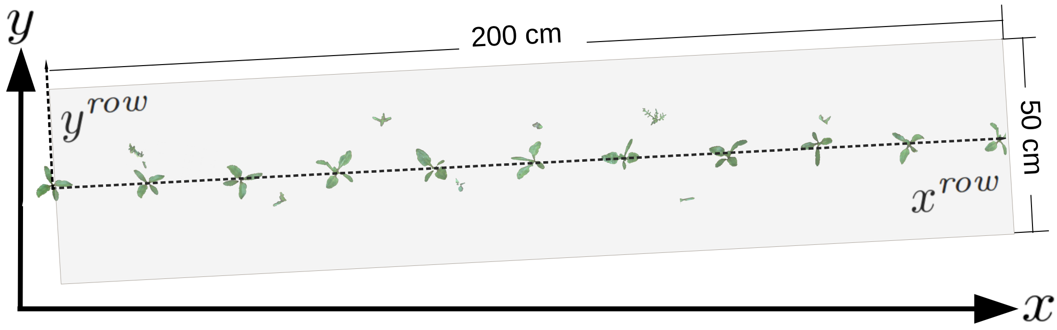


Figure 4.12: Local map of the segmented objects $\mathcal{O}$. The dashed line depicts the estimated crop row defining the coordinate system to compute $\mathcal{D}$.

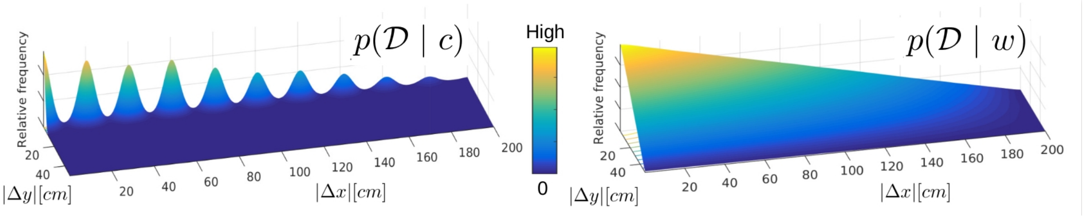


Figure 4.13: Plant arrangement model according to Equation (4.19) represented by the probability distributions of intra-class coordinates differences $\mathcal{D}$ measured between plants in a coordinate system, which is aligned to the actual crop row. Left: Probability distribution $p(\mathcal{D} \mid c)$ for sugar beets learned from data (real distribution learned after approx. 20 m of traveling on a field in Bonn). Right: Probability distribution $p(\mathcal{D} \mid w)$ for weeds obtained under the assumption that weeds spatially follow a uniform distribution in object space. The shape of $p(\mathcal{D} \mid w)$ is caused by the computation of $\mathcal{D}$ in a finite space leading to smaller probabilities for the observation of large coordinate differences.

Crop Row Detection We perform the crop row detection using a Hough transform, searching for the first voted line given the actual crop plants in the local map. Finally, we optimize the result using a least-square estimator for line estimation by considering the supporters of the detected line by the Hough transform.

Update of the Plant Arrangement We compute $\mathcal{D}$ between plant objects $\mathcal{O}_c$ or keypoints $\mathcal{K}_c$ and update our model $p(\mathcal{D} \mid c)$ for the crop plants by accumulating a 2-dimensional histogram of $\mathcal{D}$. This accumulation can be done, as the used coordinate differences are not tied to an external coordinate system. As we have only a limited amount of data, we smooth $p(\mathcal{D} \mid c)$ using a Gaussian kernel. For the weed class, we obtain the distribution $p(\mathcal{D} \mid w)$ by assuming a uniform spatial distribution of weed objects within the local map. Figure 4.13 depicts the learned arrangement model $p(\mathcal{D} \mid c)$ for crop plants as well as the assumed $p(\mathcal{D} \mid w)$ for weeds.

4.4.3 Predictions of the Geometric Classifier

We compute the coordinate differences $\mathcal{D}$ from a new object $\mathcal{O}$ or keypoint $\mathcal{K}$ to already classified plants in the local map and employ Bayes rule to obtain the probability

$$p(c \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid c)\, p(c)}{\sum_{\omega} p(\mathcal{D} \mid \omega)\, p(\omega)}, \tag{4.22}$$

for an object $\mathcal{O}$ or keypoint $\mathcal{K}$ belonging to the crop class. The distribution $p(c \mid \mathcal{D})$ reflects the output of the geometric classifier. In addition to the update of the plant arrangement model, we also update the priors $p(c)$ and $p(w)$ according to the observed class-wise occurrence counts during operation. In the training phase, we compute $p(c)$ and $p(w)$ once based on the total amount of crop plants and weeds present in the training data.

4.4.4 Combining Visual and Geometric Classifiers

This section describes our RF-GC approach that integrates the geometric information into the visual plant classification system. We combine the visual and the geometric classifier to compute a joint classification of the crop plants and weeds and to achieve an online adaption of the visual classifier to match better with the actual distribution of the visual features.

Joint Classification It is safe to assume that the features used by the visual and geometric classifiers are independent of each other. If the training labels used by both classifiers are partially the same, the resulting classifiers, however, may not necessarily be independent. We make the independence assumption and compute the class label ω^* for an object $\mathcal{O}$ by maximizing the product of both distributions:

$$\omega^* = \underset{\omega}{\operatorname{argmax}}\, p(\omega \mid \Phi(\boldsymbol{V}, \Theta))\, p(\omega \mid \mathcal{D}). \tag{4.23}$$

By combining the two outputs of the classifiers, the entire system works even if only one of the two classifiers provides an output. In this case, we consider a uniform distribution for the classifier with no output. Thus, ω^* turns into the response of the other classifier.

Table 4.2: Actions for the semi-supervised RF-GC approach.

No.	Vision	Geometry	Prediction	Action
1	confident	confident	agree	nothing
2	uncertain	uncertain	(any)	nothing
3	uncertain	confident	(any)	add label to vision
4	confident	confident	contradict	check crop row
5	confident	uncertain	(any)	check crop row

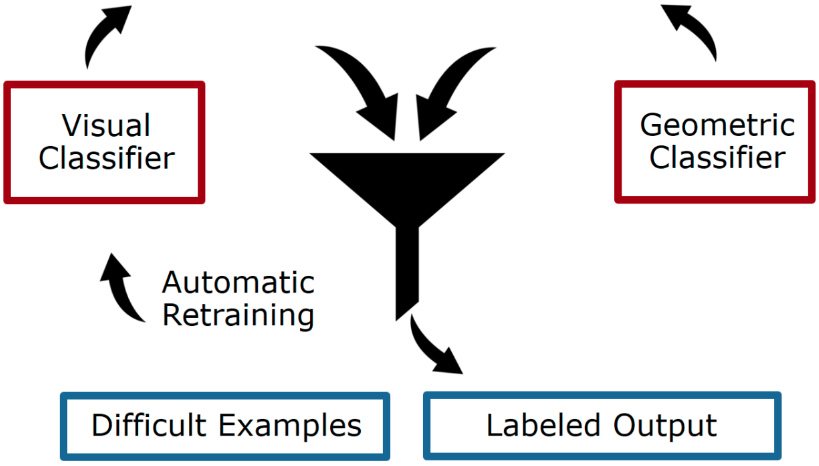


Figure 4.14: Online adaptation of the visual classifier.

Semi-Supervised Learning by Exploiting Crop Arrangement The goal of the semi-supervised approach in RF-GC is first to exploit the predictions of the geometric classifier that are not affected by the potential change in visual appearance of the plantation in order to generate new training data for the visual classifier and second to use the predictions of the visual classifier to identify errors in the crop row estimation. Concerning the predictions provided by both classifiers, we perform different actions such as adding predictions of the geometric classifier as ground truth labels for the visual classifier or perform a check of the estimated crop row. We list the considered actions in Table 4.2.

In cases No. 1 and 2, we perform no action. In case No. 3, we add the label provided by the geometric classifier to the visual training data to adapt the system to the field-specific visual feature distribution in the next retraining step. For the cases No. 4 and No. 5, we see the following reasons to check the crop row: (i) if the appearance of the plants changed substantially so that the visual classifier fails while assuming to provide confident results. From our experience, this is the most common failure case. (ii) The crop row detection is wrong. In practice, crop row detection can fail if crop plants are not present for a longer period, e.g., due to errors during sowing or because the robot moved outside the crop row so that no plants are visible (in combination with failure of the odometry system). To check which classifier to trust in such situations, we estimate two new plant arrangement models, one based on the current visual classifier and one only from the geometric one. In both cases, we use the plant information from the last 2 m of travel. This yields the models $p(\mathcal{D} \mid c_{vis})$ for the visual and $p(\mathcal{D} \mid c_{geo})$ for the geometric classifier as well as two independent estimates of the location of the crop row. Then, we compare the distributions $p(\mathcal{D} \mid c_{vis})$ and $p(\mathcal{D} \mid c_{geo})$ to our currently used relative plant arrangement model $p(\mathcal{D} \mid c)$. The comparison is made using the Kullback-Leibler divergence, a general measure for the similarity of distributions. We trust the model, which has a smaller distance under the Kullback-Leibler divergence. If the geometric classifier is assumed to be correct, we follow case No. 3. Otherwise, we use the crop row estimated by the visual classifier and proceed with the existing plant arrangement model.

Online Adaptation of the Visual Classifier The main goal of the online adaption of the visual classifier is to optimize its model to obtain high-quality classification outputs for the vegetation objects currently being observed in the field. The random forest framework offers two options: Either we retrain individual trees, or we gradually replace individual trees in the random forest as new training data arrives.

Here, we explicitly utilize the newly gathered training data generated during the operation in the actual field environment to achieve the adaption of the visual classifier model to the current distribution of features, see Table 4.2. Figure 4.14 illustrates the data flow for the semi-supervised online learning approach. We construct a new tree after having obtained a given amount of training samples generated by the geometric classifier and combine them with randomly chosen training samples obtained during the whole operation in the actual field. This procedure leads to a field-specific adaptation of

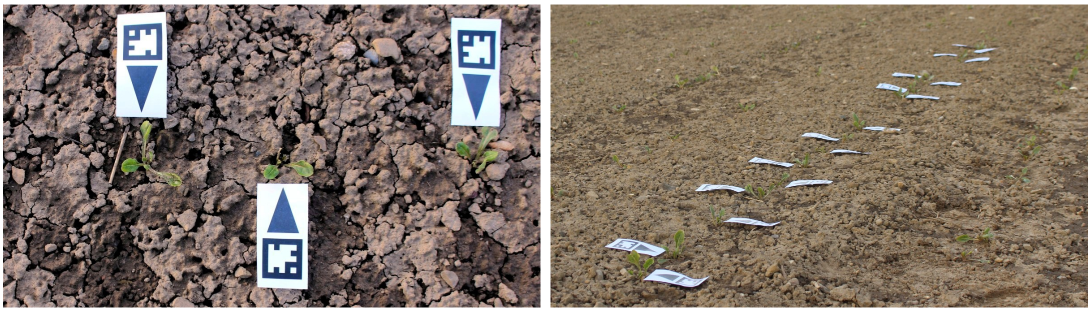


Figure 4.15: Labeling of the crop plants (sugar beets) with markers placed next to the plant. We find the markers in the images and assign the label "crop" to detected vegetation based on a distance threshold, all other vegetation is considered to belong to the "weed" class.

the random forest and thus generally leads to a better classification for this environment. In this way, for each field, a new adaptation is possible, starting with an existing classifier (typically learned over multiple fields).

Initialization of the Geometric Classifier To produce predictions with the geometric classifier, the plant arrangement model $p(\mathcal{D} \mid c)$ needs to be properly initialized, i.e., it must reflect the actual plant arrangement in the field. We propose two different procedures for the initialization.

The first procedure is the initialization of $p(\mathcal{D} \mid c)$ through predictions obtained by the visual classifier. Here, the robot first traverses around 2 m of a row, and the visual classifier detects the crop plants and weeds along the traversed field surface. These predictions are then used to compute $p(\mathcal{D} \mid c)$. After the initialization phase, the geometric classifier also can produce predictions for vegetation objects and supports the visual classifier through our proposed semi-supervised approach. The benefit of this initialization procedure is that no labeled data is necessary for its execution. A serious disadvantage is that a wrongly initialized plant arrangement model can cause further falsely classified crop plants and weeds.

The second procedure is based on in-field labeling using artificial markers. We target an in-field labeling effort of approx. 1 minute for a human operator and do not consider any pre-trained classifier. We achieve this 1-minute labeling effort by placing printed markers next to a set of crop plants at the beginning of the row. We can place around 10-15 markers within a minute, which corresponds to approx. 2-3 m of sugar beets along a row, see Figure 4.15. The placement of the markers is the only labeling effort that we use. Then, we assign the labels ω_c and ω_w concerning the distance from markers to the detected vegetation within the local map. Based on this information, we can initialize the plant arrangement model and start training the visual classifier from scratch or the adaption of a pre-trained model.

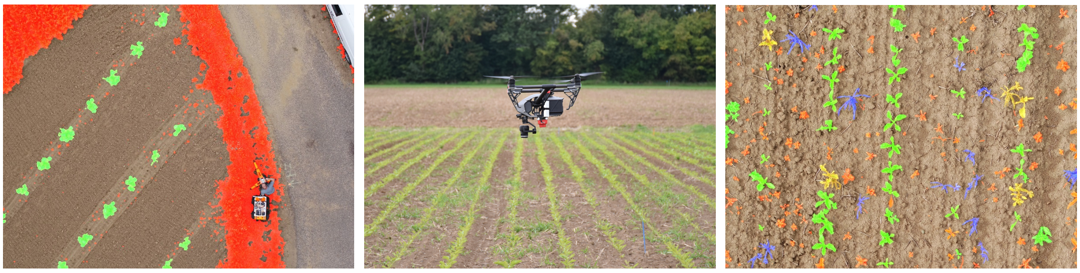


Figure 4.16: From left to right: analyzed image by our approach classifying pumpkin (green) and weed (red). UAV operation in a sugar beet field. Multi-class prediction of an image of a sugar beet field considering sugar beet (green), weed (red), grass (blue) and mixed (orange).

4.5 Adaption to UAVs

Using UAV data instead of data recorded with a UGV is more challenging, as the imagery is naturally exposed to varying lighting conditions and different scales in terms of the ground sampling distance. Figure 4.16 illustrates two exemplary classification results and one of the UAVs used for data acquisition. Thus, UGV-based systems can typically exploit more assumptions about the data. UGVs are capable of controlled illumination as the sun-light can be shielded, and artificial light sources can be applied.

Our main adaptation for the UAV-based plant classification is an extension of the RF-CAS approach by adding more relevant handcrafted features exploiting specific characteristics of UAV images. We call our UAV-based plant classification approach RF-UAV. Note that for UAV images, we solely rely on $\mathbf{I}_{RGB}$ images as input to the network. The desired output is also given by the plant mask $\mathbf{I}_{\omega}$, whereas the desired classes vary regarding the application and used dataset in our experimental evaluation in Chapter 6. For the value crop, we consider sugar beets, peppermint, strawberries, and pumpkin. For weeds, we either consider a general weed class as for the UGV case, or we explicitly classify different weed species.

In addition to the features described in Section 4.3.3, we consider further geometric features exploiting the field geometry. Usually, UAV images of fields capture larger areas compared to the ones captured by UGVs. Thus, they observe a sufficient number of crop plants within a single image to perform a row detection and to measure spatial relationships among multiple individual plants. We investigate additional geometric features to exploit the fact that crops mostly have a regular spatial distribution without explicitly specifying it. Note that weeds may also appear spatially in a systematic fashion, e.g., in spots or frequently in border regions of the field. First, we perform a line set detection to find parallel crop rows and use distances from potential rows to $\mathcal{O}$ and $\mathcal{K}$ as a feature for the Random Forest classifier. Second, we compute distributions based on distances and angles in the local neighborhood around objects and keypoints and extract statistics from it to use them as additional features.

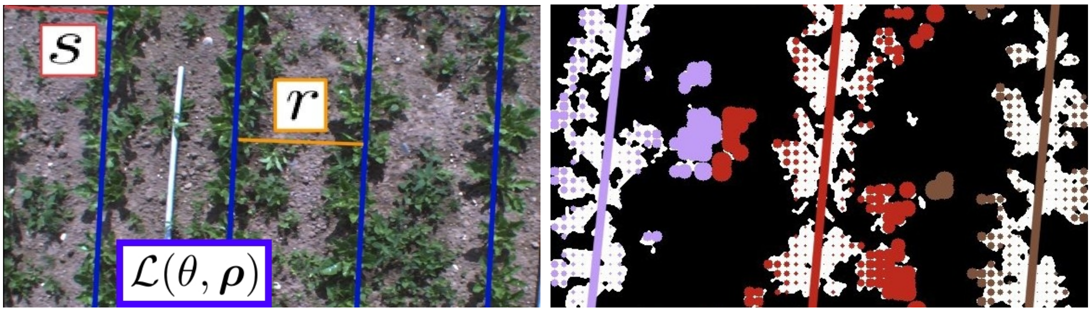


Figure 4.17: Left: Result of the line set detection. s (red) refers to the distance of the first line in the set $\mathcal{L}(\theta, \boldsymbol{\rho})$ to the origin of the image frame. r (orange) refers to the inter-row space. Right: visual illustration of the line model feature $\mathcal{V}_{line}$ for the keypoint-based approach. The different colors refer to different lines of the detected line set. The radius of a keypoint encodes the value of $\mathcal{V}_{line}$. Weeds located within the inter-row space get a higher value in $\mathcal{V}_{line}$.

4.5.1 Line Feature for Crop Rows

In most agricultural field environments, the plants are arranged in rows, which share a constant inter-row space, i.e., the distance between two neighboring crop rows. The main goal of the line feature

$$\mathcal{V}_{line} = \frac{d}{r}, \tag{4.24}$$

is to exploit the distance d of an object $\mathcal{O}$ or keypoint $\mathcal{K}$ to a crop row. We normalize d by the inter-row space r and use $\mathcal{V}_{line}$ as an additional feature for the classifier. The values d and r are measured in pixels and can be directly obtained in image space. From a mathematical point of view, crop rows can be represented as a finite set of parallel lines.

$$\mathcal{L}(\theta, \boldsymbol{\rho}) = \{l_1(\theta, \rho_1), \ldots, l_I(\theta, \rho_I)\}_{i=1}^{I}, \tag{4.25}$$

where, θ refers to the orientation of the line set and $\boldsymbol{\rho}$ are the distances from each line l_i to the origin of the image frame.

Figure 4.17 depicts an exemplary result of a detected line set and illustrates the concept of our line-based feature. We introduce the constraint that ρ_i are equidistant to exploit the fact that the inter-row space of crop rows is constant. Note that we do not make any assumptions about the size of r, i.e., the inter-row space. To detect the set $\mathcal{L}(\theta, \boldsymbol{\rho})$ of parallel lines, we employ the Hough transform on the vegetation mask $\mathbf{I}_{MASK}$. This Hough space accumulates the number of votes $v_{\rho,\theta}$, i.e., the number of corresponding vegetation pixels for a given line with the parameters θ and ρ. To compute $\mathcal{L}(\theta, \boldsymbol{\rho})$, we analyze the Hough space and perform the following three steps.

Step 1: Estimating the main direction of the crop rows We compute the main direction $\theta_{\mathcal{L}}$ of the vegetation in an image in order to estimate the direction of the crop rows. This direction can be estimated by considering the votes for parallel lines in Hough space. Here, we follow an approach similar to the one proposed by Midtiby and Rasmussen [94]. To obtain $\theta_{\mathcal{L}}$, they compute the response

$$E(\theta) = \sum_{\rho} v_{\rho,\theta}^2, \tag{4.26}$$

for each direction and select the maximum $E(\theta)$. The term $v_{\rho,\theta}$ refers to the number of votes for a particular line with the parameters θ, ρ. In contrast to [94], we do not only select the maximum of $E(\theta)$ but consider the N best values for $E(\theta)$ to evaluate the N best-voted directions in Hough space given the vegetation in the subsequent steps. In our implementation, we use $N = 15$. We consider the 15 best main directions supported by the vegetation in order to handle scenarios with large amounts of weed. Tests under high weed pressure show that the maximum response is not always the correct choice, as many weed plants may lead to more votes for false detection of the rows.

Step 2: Estimating the crop rows as sets of parallel lines Given the N best-voted orientations of possible line sets from Step 1, we want to estimate in which direction we find the best set of parallel lines with equidistant spacing.

We search for an unknown but constant spacing r between neighboring lines as well as the offset s of the first potential crop row in image space, see Figure 4.17 for an illustration. Thus, we search for the maximum response of

$$E(\theta, r, s, L_r) = P + \sum_{r=1}^{R} \sum_{s=0}^{r} \sum_{l=0}^{L_r - 1} v_{(s+l\,r),\theta}, \tag{4.27}$$

with the penalty term

$$P = -L_r\, \bar{v}_{\theta}, \tag{4.28}$$

by varying the size of r and s. The term L_r refers to the number of lines that intersect with in the image for a given r. The penalty term P is an additional cost term that is introduced for each line of the set in order to penalize an increasing number of lines. Here, $\bar{v}_{\theta}$ is the mean response over the column corresponding to $\theta_{\mathcal{L}}$ in the Hough space. This leads to the effect that E, according to Equation (4.27), increases for lines, which have a better response $v_{s+l\,r,\theta} > \bar{v}_{\theta}$ and vice versa decreases if the response is lower. The maximum response according to Equation (4.27) provides the best voted line set, which has a constant inter-row space.

Step 3: Refitting the best line set to the data Crops are commonly sown out in fixed assemblages of a certain number of rows, called plots. It can happen that the inter-row space between plots differs slightly due to the limited position accuracy of the sowing machine. To overcome this, we finally fit each line of the best set obtained from step 2 to the data by using a robust estimator based on a Huber kernel and obtain a robust estimate $\mathcal{L}(\theta, \boldsymbol{\rho})$ for the crop rows.

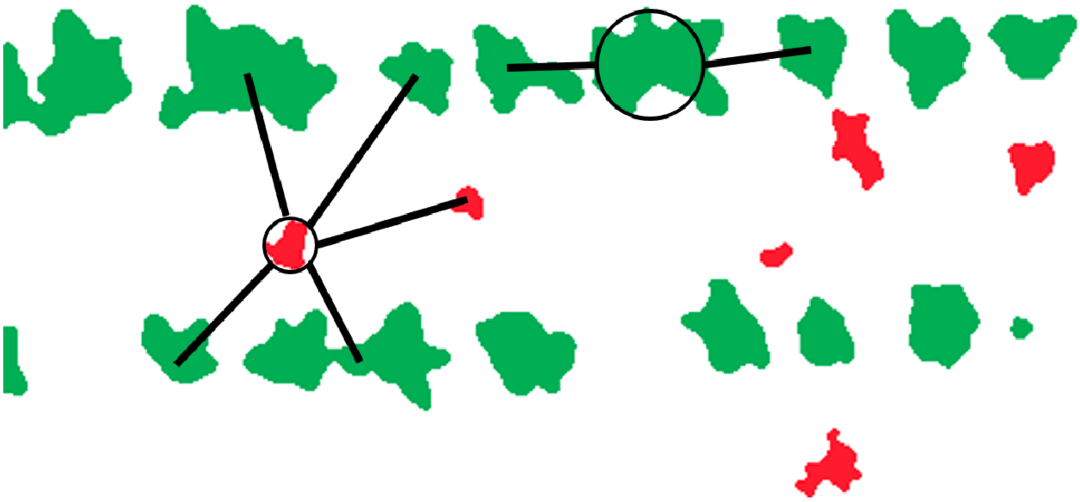


Figure 4.18: To describe spatial relationships among individual plants, we compute spatial relationship features for every object and every keypoint in an image considering distances and azimuths from a query object $\mathcal{O}_q$ or keypoint $\mathcal{K}_q$ to all other nearby objects or keypoints.

4.5.2 Spatial Relationship Features

In order to describe spatial relationships among individual plants, we compute spatial relationship features for every object and every keypoint in an image. First, we compute the distances and azimuths from a query object $\mathcal{O}_q$ or keypoint $\mathcal{K}_q$ to all other nearby objects or keypoints in world coordinates (which requires knowing the flying altitude of the UAV). We compute the differences between the measured distances of the query object and its neighbors. Through this, we obtain a distribution in the form of a histogram. Similarly, we obtain the distribution over angles from the observed azimuths. From these distributions, we compute common statistical qualities, such as min, max, range, mean, standard deviation, median, skewness, kurtosis, and entropy (see Table 4.1), and use them as features for the classifier. In addition to that, we count the number of vegetation objects $\mathcal{O}$ or keypoints $\mathcal{K}$ in their neighborhood in object space. Figure 4.18 illustrates the considered neighbors for the extraction of the spatial relationship features for one crop object (green) and one weed object (red). We limit the considered neighbors by a certain radius for two reasons. First, to grasp information about the local arrangement of the plants and weeds. Second, to control the computational efforts. Both the spatial relation features as well as the line features allow for encoding additional geometric properties and, in this way, to improve the random forest classifier used to make the actual decision.

4.6 Summary

In this chapter, we presented different approaches to the random forest-based plant classification. All these approaches follow the main pipeline shown in Figure 4.1 and use handcrafted features that encode visual information. The first two approaches, RF-KP and RF-OBJ, address the feature extraction for the classification problem in different ways. The keypoint-based approach RF-KP extracts local features for keypoints and classifies the area around each keypoint. The object-based approach RF-OBJ extracts features for each object- or segment-based and classifies all pixels within a vegetation segment. Please note that for RF-KP, we do not claim a contribution in this thesis, as we developed this approach with the context of the master thesis by Lottes [81].

Next, we propose a way to link RF-OBJ and RF-KP within our approach RF-CAS, which combines the object-based and keypoint-based feature extraction and classification in a cascade and exploits their respective advantage and even compensates for their respective disadvantages.

Furthermore, we propose a probabilistic model encoding the spatial arrangement of the crop plants and weeds in the field using coordinate differences between plants. We integrate this model into our next contributed approach RF-GC, which combines the visual random forest classifier RF-CAS with a geometric Bayesian classifier in a semi-supervised way.

Finally, we extend our proposed RF-CAS approach to also appropriately process UAV images, i.e., RF-UAV. Here, we consider further geometric features exploiting the field geometry in terms of crop row information and spatial relationships among multiple individual plants.

Chapter 5

Plant Classification Using Fully Convolutional Neural Networks

THE main objective of this thesis is the development of innovative vision-based plant classification systems for agricultural robots, allowing the robots to identify the value crop and distinguish it from weeds or even different weed species. In this chapter, we introduce our second series of plant classification systems enabling UGVs and UAVs to perceive crop plants and weeds in agricultural field environments.

It has been shown that fully convolutional neural networks for pixel-wise classification, i.e., semantic segmentation tasks achieve superior performance for a large number of different applications [5, 55, 108, 126]. In this chapter, we introduce our second series of plant classification systems enabling UGVs and UAVs to perceive crop plants and weeds in agricultural field environments. All classification systems, we present in this chapter, use fully convolutional neural networks as their core machine-learning model. The classification systems operate with RGB images ($\mathbf{I}_{RGB}$) as well as with 4-channel images, which consist of $\mathbf{I}_{RGB}$ plus additional near-infrared measurements per pixel, i.e., $\mathbf{I}_{NIR}$. The goal is to map the image input into a label map that encodes a class label for every pixel.

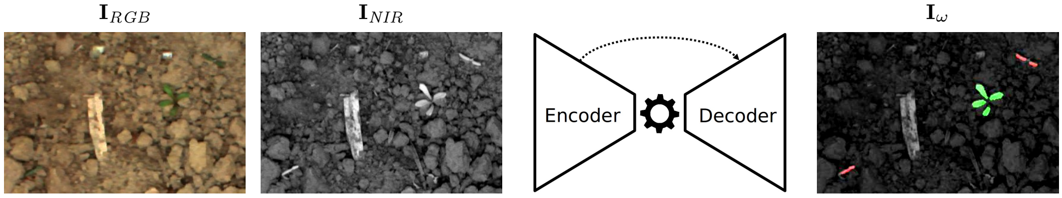


Figure 5.1: Plant classification pipeline based on fully convolutional neural networks. From left to right: $\mathbf{I}_{RGB}$, $\mathbf{I}_{NIR}$, fully convolutional neural network-based classification model, predicted label mask $\mathbf{I}_{\omega}$. In contrast to the random forest-based approaches described in Chapter 4, the fully convolutional neural network-based approaches do not perform a preliminary vegetation classification step.

We illustrate the key processing steps of the fully convolutional neural network-based approaches in Figure 5.1. First, we feed the $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images into the network. In contrast to the proposed approaches in Chapter 4, fully convolutional neural networks neither rely on the design of handcrafted features nor require a preliminary vegetation detection step before the actual plant classification. Handcrafted features, however, can limit the capacity, i.e., the representational power of the classification model. Fully convolutional networks, in contrast, operate in an end-to-end manner. They learn the features and perform the classification simultaneously.

The fully convolutional neural network-based plant classification systems in this chapter enable agricultural field robots to recognize and locate the crop plants and weeds. Through this, field robots can perform automated in-field treatments such as selective and plant-specific treatments. One of the key components of fully convolutional neural networks is its architectural design. In our experimental Chapter 6, we analyze the plant classification performance obtained by different architectures such as Resnet-34 [52], DarkNet [124], ErfNet [125], MobileNet-V2 [132], and DeepLab-V3+ [22]. The architectures of these related works are typically designed for complex classification problems with up to a thousand different classes. They contain several million free parameters that are learned during training.

Our experiments in Chapter 6 suggest that DenseNet-based architectures based on the work of Huang *et al.* [55] and Jegou *et al.* [58] provide the best performance and a faster convergence time in the training phase. Thus, we design an architecture based on architectural building blocks from DenseNets and modify it for the plant classification task tasks at hand. We design our network architecture, keeping in mind that the classification needs to provide results online such that an actuator can directly act upon the incoming information while the robot traverses the field. The plant classification tasks that we consider in this thesis range from a minimum of three classes for the crop-weed classification to a maximum of seven classes for the crop-dicot-grass classification with joint crop-dicot stem detection. Thus, we design lightweight networks explicitly for the plant classification task.

5.1 Different Fully Convolutional Neural Network-Based Classification Systems

Throughout this chapter, we propose five different variants for the fully convolutional neural network-based plant classification. We refer to a fully convolutional neural network-based approach with FCN-*, where FCN stands for the fully convolutional neural network. The postfix is a placeholder for abbreviation referring to the respective variant. All approaches follow the main pipeline shown in Figure 5.1. In the course of this chapter, we will also introduce modifications of these approaches that exploit additional geometric features about the local arrangement of plants in the field to distinguish plants and weeds.

Our first approach is called FCN. In Section 5.2.1, we design a lightweight encoder-decoder structured network architecture that performs a pixel-wise classification considering the classes $\omega^{cws} \in \{\omega_{\mathsf{c}}, \omega_{\mathsf{W}}, \omega_s\}$ for crop, weed, and soil. The network architecture of the encoder and the decoder presented therein represents the basic architectural design for all approaches presented in this chapter.

Subsequently, we present the FCN-UAV approach that we explicitly design for processing UAV images in Section 5.2.2. UAV images differ from UGV images mainly in terms of the ground resolution and the camera's footprint in object space. The architecture of FCN-UAV follows that of our FCN approach, but we adjust the receptive field to use a larger area in the image for the classification of a pixel. In this way, we allow the FCN-UAV approach to implicitly incorporate information about the relative arrangement of plants in the image in the feature extraction.

A further goal of this work is to use robots to control different weeds with different treatments. For this purpose, the field robot has to be able to differentiate not only between plants and weeds but also between weeds that should be treated differently. High-precision interventions, such as precise mechanical and laser-based weeding, are most effective when applied to the stem locations of small weeds. Selectively spraying agrochemicals over their entire leaf area, however, is still the most effective approach to treat big weeds and generally grass weeds. Therefore, we introduce FCN-STEM in Section 5.3. FCN-STEM is a single model for jointly determining both, the exact stem location of dicotyl weeds and plant as well as pixel-wise plant classification considering the classes $\omega^{cdgs} \in \{\omega_{\mathsf{c}}, \omega_d, \omega_g, \omega_s\}$ for crop, dicotyl weeds, grass weeds, and soil. In addition, we present in Section 5.3.3 our FCN-STEM approach in the context of plant counting. Stem detection turns out to be particularly suitable for the identification of single plants.

Next, we propose a way to exploit additional geometric prior information about the local arrangement of the plants in the field to improve the performance of the classification systems in terms of performance and generalization capabilities to new and changing field conditions. We integrate this information by analyzing image sequences that cover a local strip of the field surface and thus implicitly carry the information about the plant arrangement. Section 5.4.2 describes one of our main contributions, i.e., the sequential module, which is a subnetwork that analyzes visual features of consecutive images from a sequence and extracts spatio-temporal features that encode the field geometry. The integration of the sequential module into our FCN approach leads to our proposed novel FCN-SEQ approach, which we describe in Section 5.4.3. FCN-SEQ is an end-to-end trainable network that provides a pixel-wise classification of plants and weeds exploiting image sequences.

In Section 5.4.4, we also integrate the sequential module into our FCN-STEM approach and propose our novel FCN-SEQ-STEM approach that jointly predicts stem locations and performs a pixel-wise plant classification exploiting spatio-temporal features. Throughout our experimental evaluation, we demonstrate the ability of the sequential module to extract spatio-temporal features encoding the relative arrangement of the plants. Furthermore, we empirically show that our sequential approaches provide superior classification and generalization performance compared to the approaches that process single images independently.

We presented the FCN as well as the FCN-STEM approach in [82], whereas we published the FCN-SEQ approach in [84], the FCN-SEQ-STEM approach in [83], and the deployment of our FCN-UAV approach for UAV-based crop monitoring applications in [85].

All proposed approaches for UGVs achieve a sufficient runtime for the processing of the classification results that are required for online in-field operations such as selective spraying or mechanical weed control. We implement the inference for all proposed variants of the fully convolutional neural network-based plant classification systems as ROS modules and evaluate them on different real field robots, see Chapter 3. For training and inference, we use the TensorFlow library [2].

5.2 Fully Convolutional Neural Network-Based Plant Classification: FCN

This section describes our base, fully convolutional neural network model for pixel-wise plant classification on single images, which we call FCN in the remainder of this thesis. The main objective is to provide a pixel-wise classification of the input images $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$, where we either use solely $\mathbf{I}_{RGB}$ or $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ as input to the network. In the $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ case, we concatenate the images along their channel axis such that we obtain a 4-channel image as input. The desired output is given by the plant mask $\mathbf{I}_{\omega^{cws}}$ considering the classes $\omega^{cws} \in \{\omega_{\mathsf{c}}, \omega_{\mathsf{w}}, \omega_{s}\}$ for crop, weed, and background.

The principal processing pipeline executes the following key steps and is illustrated in Figure 5.1. First, we preprocess each image according to Section 4.2. Next, we feed the preprocessed images into the encoder-decoder structured architecture, which directly outputs the per-pixel probability distribution $p(\omega^{cws} \mid FCN)(i,j)$ over the desired class labels for each pixel location (i,j). Finally, we obtain the label mask $\mathbf{I}_{\omega^{cws}}$ by determining the label with the highest probability according to Equation (2.23), i.e.,

$$\mathbf{I}_{\omega^{cws}} = \operatorname*{argmax}_{\omega} p(\omega^{cws} \mid FCN)(i,j). \tag{5.1}$$

Equation (5.1) reflects the standard approach to assign the class labels based on the probabilistic output of a classifier for multi-class problems. A colored illustration of $\mathbf{I}_{\omega^{cws}}$ is given in Figure 5.1 (right).

5.2.1 Encoder-Decoder Network Architecture

Common, fully convolutional neural network architectures follow the so-called "hourglass" structure, referring to a downsampling operation of the resolution in the encoder followed by a complementary upsampling in the decoder to regain the full resolution of the input image for the pixel-wise classification. More specifically, the encoder part of the network compresses the content of the input images into a small but highly informative representation, and the decoder part of the network simultaneously reconstructs

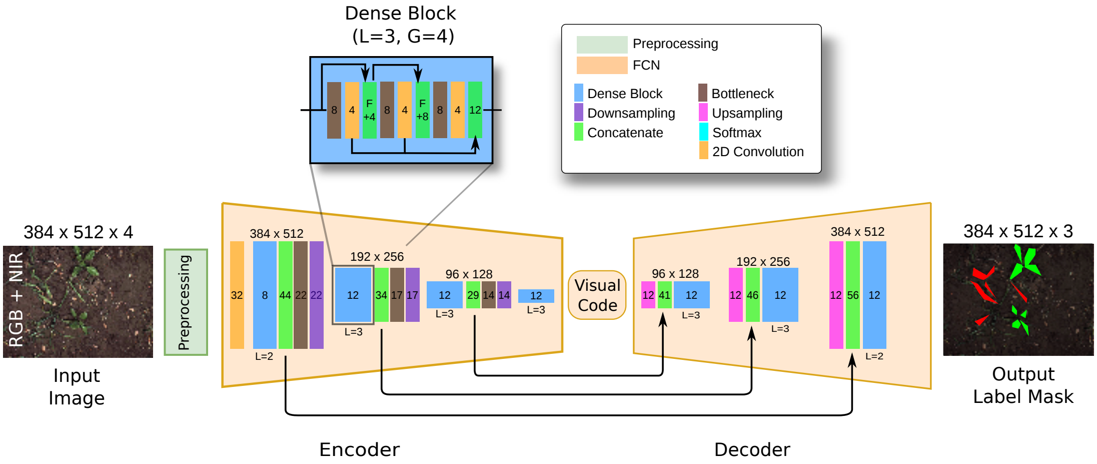


Figure 5.2: FCN architecture for pixel-wise plant classification, also called semantic segmentation. Given $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ as input, we first compute the visual code by the encoder. The visual code is then again upsampled by the decoder resulting in the visual features, which are then turned into $\mathbf{I}_{\omega^{cws}}$ considering the classes $\omega^{cws} \in \{\omega_{\mathrm{c}}, \omega_{\mathrm{w}}, \omega_{s}\}$. Dimensions are for the crop-weed classification based on the $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ input.

the spatial resolution of the input and maps it to the pixel-wise (pseudo) probabilities distribution over class labels.

As a basic building block in our encoder-decoder fully convolutional neural network, we follow the ideas of the so-called fully convolutional DenseNet [58], which combines the recently proposed densely connected CNNs [55] organized as dense blocks with fully convolutional neural networks [80]. The key idea is a dense connectivity pattern that iteratively concatenates all computed feature maps of subsequent convolutional layers in a feed-forward fashion. These "dense" connections encourage deeper layers to reuse features produced by earlier layers and additionally support the gradient flow in the backward pass. As commonly used in practice, we define our 2D convolutional layer as a composition of the following components: (i) 2D convolution, (ii) rectified linear unit (ReLU) as non-linear activation, (iii) batch normalization [141], and (iv) dropout [138]. We repeatedly apply bottleneck layers to the feature volumes and thus keep the number of feature maps small while achieving a deep architecture. Our bottleneck is a 2D convolutional layer with a $[1 \times 1]$ kernel.

A dense block is given by a stack of L subsequent 2D convolutional layers operating on feature maps with the same spatial resolution. Figure 5.2 depicts the information flow in a dense block. The input of the l^{th} 2D convolutional layer is given by a concatenation of all feature maps produced by the previous layers, whereas the output feature volume is given by the concatenation of the newly computed feature maps within the dense block. Here, all the concatenations are performed along the feature axis. The number of the produced feature maps is called the growth rate G of a dense block [55]. We consequently use two times G convolutional kernels for the bottleneck layers within a dense block. Through this, we reduce the computational cost in the subsequent 2D convolutional layers as the number of feature maps is limited to two times the growth rate.

Figure 5.2 illustrates the information flow through the fully convolutional neural network. The first layer in the encoder is a 2D convolutional layer augmenting the preprocessed 3-channel ($\mathbf{I}_{RGB}$) or 4-channel ($\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$) images using 32 [15 × 15] kernels. The following operations in the encoder are given by a recurring composition of dense blocks, bottleneck layers, and downsampling operations, where we concatenate the input of a dense block with its output feature maps. We perform the downsampling by stridden convolutions employing 2D convolutional layers with a [5 × 5] kernel and a stride of 2. All bottleneck layers between dense blocks compress the feature volumes by a learnable halving along the feature axis. In the decoder, we revert the downsampling by a stridden transposed convolution [27] with a [2 × 2] kernel and a stride of 2. To facilitate the recovery of spatial information, we concatenate feature maps produced by the dense blocks in the encoder with the corresponding feature maps produced by the learnable upsampling and feed them into a bottleneck layer followed by a dense block. In contrast to the encoder, we reduce the expansion of feature maps in the decoder by omitting the concatenation of the input of a dense block with its output.

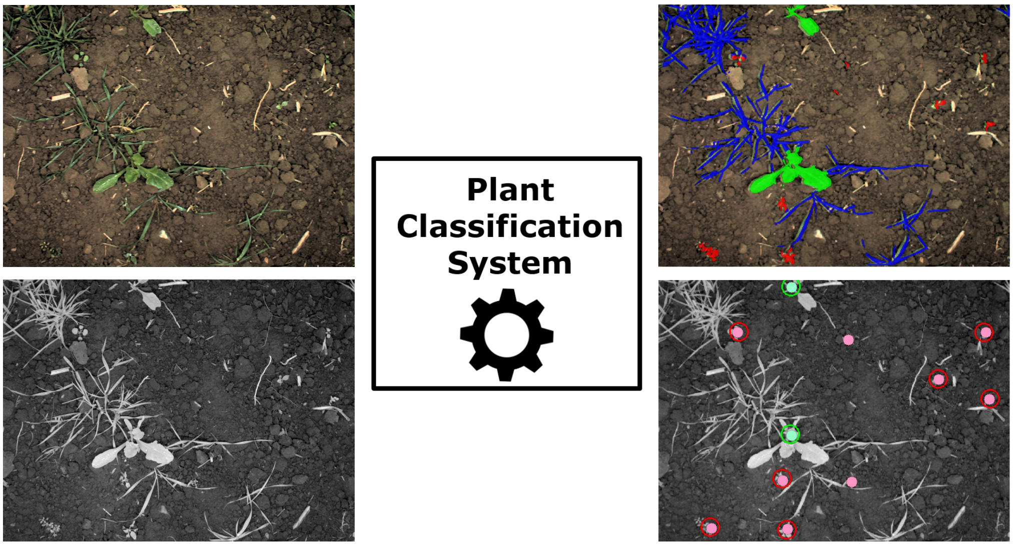
The output of the FCN network $p(\omega^{cws} \mid FCN)$ represents a pixel-wise probability distribution over the class labels. We obtain this distribution by passing the decoded features through the classification block, which is a chain of another 2D convolutional layer with eight kernels of size [15 × 15], a bottleneck convolution with C [1 × 1] kernels, where C is the number of desired class labels, and a softmax layer along the resulting depth dimension to achieve the pixel-wise probabilities for each class label. Finally, we obtain the plant label mask $\mathbf{I}_\omega$ according to Equation (2.23). Figure 5.3 (bottom left) illustrates an example for $\mathbf{I}_\omega$ projected onto the $\mathbf{I}_{RGB}$ input image.

5.2.2 FCN-UAV for UAV-Based Crop Monitoring

UAVs can be employed over an entire crop season to monitor important traits for the crop plants or to measure the spatial distribution of the crop plants and weeds. This provides a temporal dimension to the monitoring of the field, which is necessary to understand how the field status is evolving and also to know when to trigger specific field management tasks. Thus, it is crucial to have an automatic classification pipeline that monitors and analyzes the crop plants automatically and provides a report about the status of the field over time.

For the processing of UAV imagery, we propose the FCN-UAV approach. Architecturally, the network is the same as FCN, except for the depth of the encoder and decoder and the used kernel size for the convolutional layers. We add one more stage of downsampling followed by one further dense block, as described in Section 5.2.1. Through this, we enlarge the receptive field of the final feature volume of the encoder from 245 pixels to 837 pixels regarding the image input.

The UAV images typically cover a larger spatial area of the field containing sufficient information about the spatial distribution of plants and weeds. The increased receptive field enables the encoder to extract features considering a lager spatial context of the input images.



Input: RGB, NIR.

Output: Plants and stems.

Figure 5.3: Joint plant classification and stem detection pipeline. Top left: RGB image $\mathbf{I}_{RGB}$. Top right: NIR image $\mathbf{I}_{NIR}$. Bottom left: plant label mask $\mathbf{I}_{\omega}$ considering sugar beet (green), dicotyl weed (red), grass weed (blue), and background (no color) overlayed on $\mathbf{I}_{RGB}$. This result is produced by FCN-STEM approach. Bottom right: detected stems (circles) and their corresponding ground truth (filled circles) considering sugar beet (green) and dicotyl weed (red) overlayed on $\mathbf{I}_{NIR}$. This result is produced by the FCN-STEM approach

In contrast to the random forest-based RF-UAV approach for UAV-based plant classification, we do not specifically need to design handcrafted features describing the field geometry, as the FCN-UAV approach is principally able to learn this information implicitly by exploiting the larger receptive field.

Note that for UAV images, we solely rely on $\mathbf{I}_{RGB}$ images as input to the network in this thesis. The desired output is also given by the plant mask $\mathbf{I}_{\omega}$, whereas the desired classes vary regarding the application and use the dataset in our experimental evaluation in Chapter 6.

5.3 Joint Stem Detection and Plant Classification: FCN-STEM

A versatile system that aims at reducing the use of agrochemicals to the minimum necessary should not be limited to a single actuator, e.g., a selective spaying unit for all weeds. To further minimize the chemical input to a field, a weed control system may additionally be equipped with mechanical or thermal actuators. The key idea is to use the chemical-free approach wherever its possible and to rely only on spraying for cases, where mechanical or thermal methods do not work well. In this scenario, the classification system needs to provide both the stem locations and the spatial extent of the plants. The stem positions are a prerequisite for the selective, high-precision mechanical or thermal treatments, e.g., by mechanical stamping or by laser-based weeding. The provided pixel-wise label mask provides the area with more granulated treatment approaches, e.g., for selectively spraying grass weeds. The BoniRob field robots we use in this thesis are equipped with two different actuators, one for selective spraying of chemicals and another one for precise mechanical intervention by stamping.

This section describes our approach for joint plant classification and stem detection, which we call FCN-STEM. The main objective of FCN-STEM is to provide two outputs simultaneously. First, a pixel-wise classification represented by the plant mask $\mathbf{I}_{\omega^{cdgs}}$ considering the classes $\omega^{cdgs} \in \{\omega_\mathsf{c}, \omega_d, \omega_g, \omega_s\}$ for crop, dicotyl weed, grass weed, and background (mostly soil). Second, the positions of the stems for dicotyl weeds and crop plants represented by the stem mask $\mathbf{I}_{\omega^{cds}}$ considering the classes $\omega^{cds} \in \{\omega_\mathsf{c}, \omega_d, \omega_s\}$ for crop stem, dicotyl weed stem, and no stem.

One key architectural design feature of FCN-STEM is that the network shares the encoded features for classifying the stem regions as well as for the pixel-wise classification using one encoder network and two task-specific decoder networks. Thus, the output differs from the FCN approach described in the previous section as it consists of two different probability distributions over the class labels for both plant classification and stem detection.

The processing pipeline executes the following key steps and is illustrated in Figure 5.4. First, we preprocess each image according to Section 4.2. Next, we feed the preprocessed images into the one-encoder-two-decoder structured, fully convolutional neural network, which outputs a per-pixel probability distribution $p(\omega^{cdgs} \mid FCN-STEM)$ for plant classification over the desired class labels for each pixel and, furthermore, a per-pixel probability distribution $p(\omega^{cds} \mid FCN-STEM)$ representing regions within the image, which correspond to crop stems and weed stems. Analogous to the FCN approach, we obtain the label mask $\mathbf{I}_{\omega^{cdgs}}$ by determining the class label with the highest probability, according to Equation (2.23). Finally, we extract pixel-accurate stem positions, i.e. the stem mask $\mathbf{I}_{\omega^{cds}}$ from $p(\omega^{cds} \mid FCN-STEM)$, through a postprocessing step.

5.3.1 One-Encoder Two-Decoder Network Architecture

Figure 5.4 depicts the proposed architecture of our joint plant and stem detection approach FCN-STEM. The main processing steps of this approach are the preprocessing (red), the encoder (orange), the plant decoder (blue), the stem decoder (green), and the stem extraction (brown). FCN-STEM relies on the same preprocessing module described in Section 4.2 as well as on the same principle architectural building blocks as described in Section 5.2.1. We use the same encoder as for the FCN approach for the extraction of the visual code.

From the encoded and compressed visual code, we generate two separate feature volumes specialized for pixel-wise plant classification and stem detection. Thus, we have two task-specific decoders, which perform an upsampling using a stridden transpose convolution [27] with [2 × 2] kernel and a stride of 2. Both decoders also use

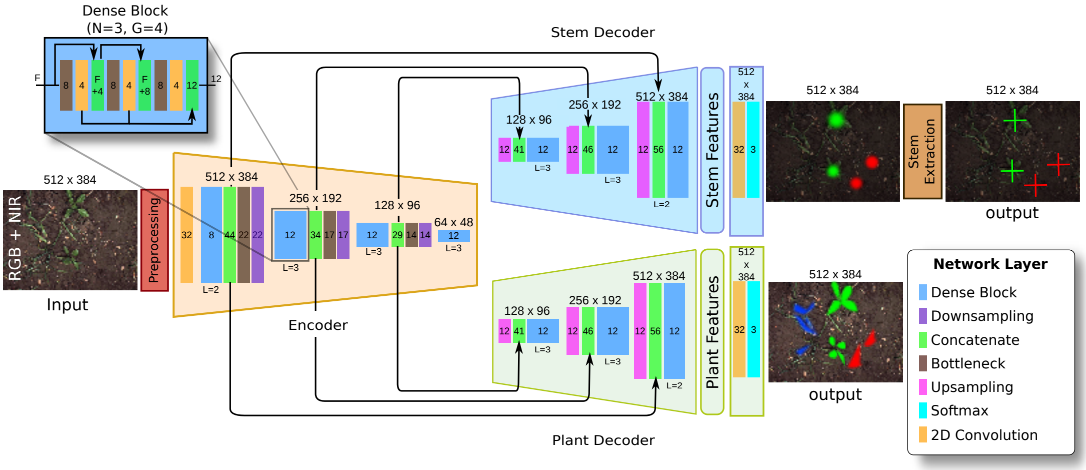


Figure 5.4: FCN-STEM architecture. We first encode the input images using the encoder and then pass the feature volumes to the task-specific decoders, the stem decoder and the plant decoder. We obtain two outputs, the plant mask $\mathbf{I}_{\omega^{cdgs}}$ considering the classes $\omega^{cdgs} \in \{\omega_\mathrm{c}, \omega_d, \omega_g, \omega_s\}$ for the pixel-wise classification of the plants and the stem mask $\mathbf{I}_{\omega^{cds}}$ considering the classes $\omega^{cds} \in \{\omega_\mathrm{c}, \omega_d, \omega_s\}$ for the segmentation crop-weed stem regions. Finally, we extract the stem positions from the stem mask in the stem extraction. Note that we denote the size of the feature map above each block of layers. Inside the layers, we show the number of output features maps.

dense blocks as their main building blocks and follow the same architectural design to produce the plant features and stem features. Moreover, both task-specific decoders use feature maps produced by the encoder through skip connections. We concatenate the corresponding feature maps sharing the same spatial resolution from the encoder before we again use dense blocks for feature computation. Skip connections from the encoder to the decoders facilitates the recovery of spatial information [5]. Finally, we transform the feature maps produced by the stem decoder and the plant decoder into the pixel-wise probability distribution over their respective class labels by a $[1 \times 1]$ convolution followed by a softmax layer.

Note that we try to predict the area of the stem instead of regressing the stem location. This is key to use the same architecture for learning plant classification and stem locations. Finally, we extract pixel-accurate stem positions from the stem mask using a postprocessing step.

Training For learning, we use a multi-task loss $\mathcal{L}$ combining the loss for the plant segmentation $\mathcal{L}_\mathrm{plant}$ and for the stem region segmentation $\mathcal{L}_\mathrm{stem}$, i.e.,

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_\mathrm{stem} + \alpha\mathcal{L}_\mathrm{plant}, \tag{5.2}$$

where we use $\alpha = 0.5$. The loss $\mathcal{L}_\mathrm{plant}$ is the weighted cross-entropy, where we penalize errors regarding the crop plants, dicotyl weeds, and grasses by a factor of 10. The loss $\mathcal{L}_\mathrm{stem}$ is based on an approximation of the intersection over union (IoU) metric, as it is more stable with imbalanced class labels [122], which is the case in our problem with under-represented stems as compared to the amount of soil. The multi-task loss

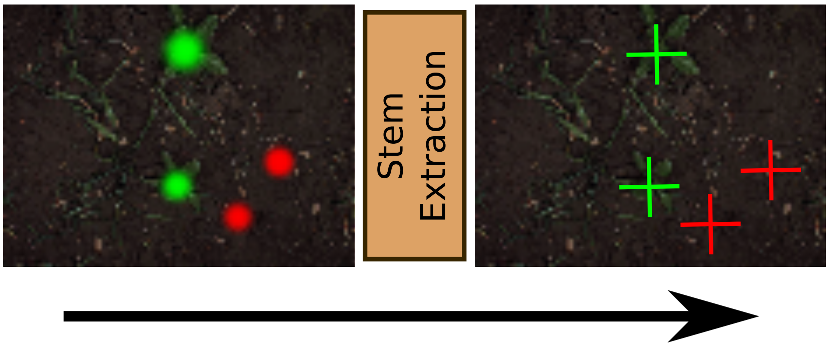


Figure 5.5: We extract the pixel-wise stem locations by computing a weighted center of mass of the stem regions predicted by the FCN-STEM network. For the weighting, we consider the predicted probabilities $p(\omega^{cds} \mid FCN-STEM)$ for each pixel belonging to a stem region.

also enables the sharing of information for learning the encoder, which can use the loss information from both decoders in the backward pass of the backpropagation. For training, we encode the stem locations as blobs with a diameter of 10 mm in object space.

5.3.2 Stem Extraction

Given the probability distribution $p(\omega^{cds} \mid FCN-STEM)$ encoding regions within the image, which correspond to crop stems and weed stems, we want to extract a well-defined stem location by a certain pixel location for the crop plants and the dicotyl weeds. To this end, we first determine $\mathbf{I}_{\omega^{cds}}$ by selecting the class with the highest label probability for each pixel. Next, we determine the connected components $\mathcal{O}^c_j$ for the crop ω_c and dicotyl weed ω_d class and compute the weighted mean $\bar{\mathbf{x}}^\omega_j$ of the pixel locations by

$$\bar{\mathbf{x}}^\omega_j = \frac{\sum_{\mathbf{x}\in\mathcal{O}^\omega_j} P(\boldsymbol{\omega} = \omega \mid \mathbf{x}) \cdot \mathbf{x}}{\sum_{\mathbf{x}\in\mathcal{O}^\omega_j} P(\boldsymbol{\omega} = \omega \mid \mathbf{x})} \quad \text{with} \quad \omega = \omega_\mathsf{c}, \omega_d. \tag{5.3}$$

The weighted means $\bar{\mathbf{x}}^\omega_j$ for class c are then the stem detections reported by our approach. Figure 5.3 (bottom right) illustrates an example for the detected stems projected onto the $\mathbf{I}_{NIR}$ input image.

5.3.3 FCN-STEM for UAV-Based Plant Counting

Plant counting represents a prevalent task for farmers and breeders. The exact knowledge about the number of emerged plants is a necessary trait to estimate. In the early season, this information helps to assess the seed quality as well as the overall plant performance. During the late season, this information indicates the expected yield and also contributes to the planning of the harvest. Nowadays, growers have had to count plants by hand by sampling certain areas in the field. This approach is imprecise because it assumes a homogeneous distribution of the plants, which is difficult to

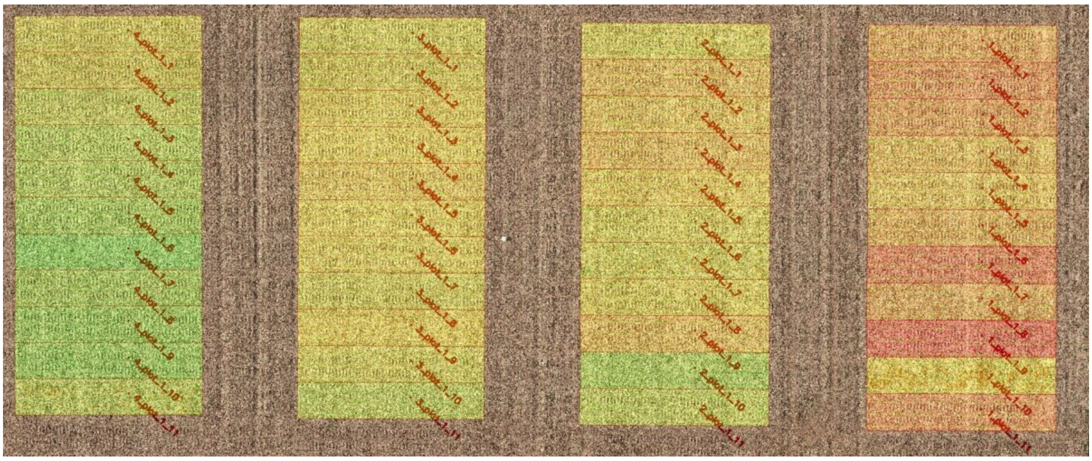


Figure 5.6: Overview of the GOETT-UAV-19 dataset containing micro plots. The colors refer to the average size of the sugar beets. Green refers to small and red refers to big plants regarding the average growth stage.

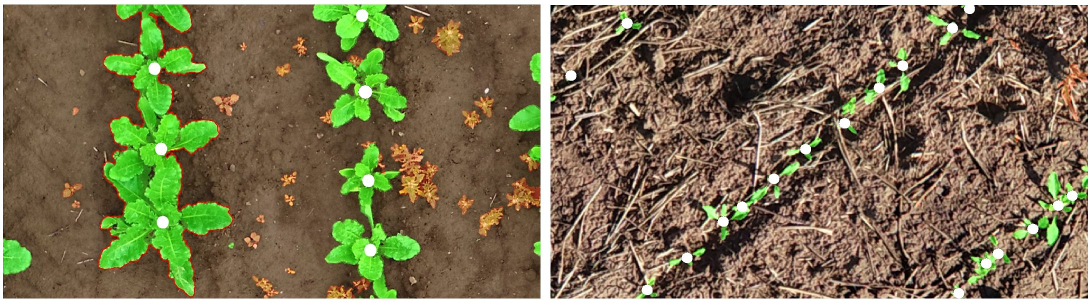


Figure 5.7: Illustration of difficult conditions for counting plants using vision-based classification approaches. Left: Mutually overlapping sugar beets. Right: Due to narrow seeding, the sugar beets overlap early after the emergence phase. In addition, individual and contiguous plants are separated by straw in the image space.

guarantee, especially for large farms. Even more laborious is the task of plant-counting within a plant-breeding scenario. Here, different crop varieties are arranged in micro plots, such as depicted in Figure 5.6. For each plot, the exact number of emerged plants has to be measured to compare the emergence performance of the different varieties.

In practice, scenarios occur in which the solution to the plant counting problem can only be solved to a limited extent with our FCN approach, see Figure 5.7. Mutually overlapping crop plants often occur when the plants have reached a particular growth stage and fill the area within and between the rows. In crop production, this row closure usually occurs a few weeks after sowing. Nevertheless, the knowledge about the exact number of plants in this growth phase is still valuable, as it helps the practitioners to predict the yield. In the case of plant breeding, overlapping plants can occur even earlier, as they are usually sown using the so-called narrow sowing method. Here, the plants share only a fraction of the usual distance in the crop row between each other. Moreover, single plants can be fragmented by straw or weeds. Since the FCN approach performs a pixel-wise classification, we cannot explicitly recognize the individual plants

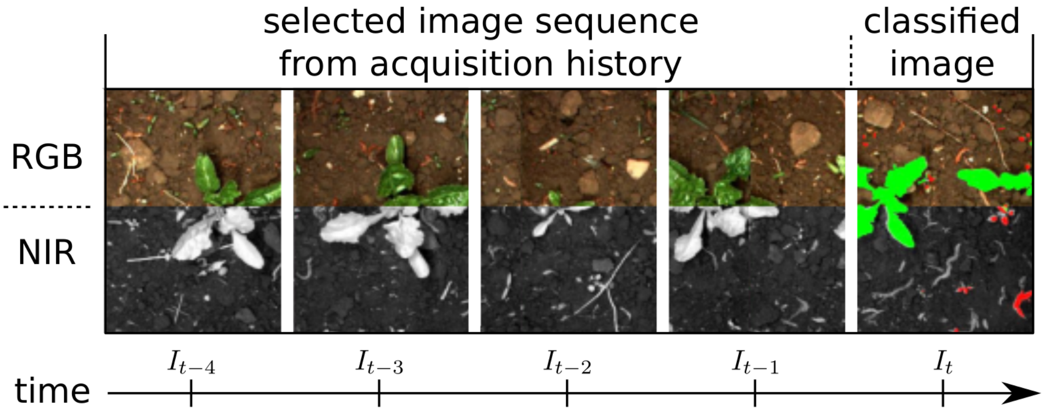


Figure 5.8: Our sequential approaches analyze image sequences of local field strips. Given a sequence of images, $I_{t-4}, \dots, I_t$, at current timestamp t, FCN-SEQ and FCN-SEQ-STEM determine a pixel-wise classification (green for crop, red for weed, blue for grass weed) and FCN-SEQ-STEM jointly predicts the stem positions (crosses) of crop plants and dicotyl weeds, as shown for I_t.

by a subsequent connected component analysis as we would either underestimate overlapping plants or overestimate fragmented plants in image-space. From this point of view, the FCN approach is not suitable for counting plants.

However, the FCN-STEM approach can identify the stem area of the plants. This area is mostly not covered by other plants and is, therefore, a sufficient proxy for the stand count of plants during the entire season. In our experiment in Section 6.10, we show that our FCN-STEM is suitable for counting plants under the aforementioned conditions.

5.4 Exploiting Plant Arrangement for Generalizing to New Environments

One major development goal in this thesis is to minimize the performance loss in plant classification when deploying farming robots for weed control equipped with a previously trained classifier in new and unseen field environments. To achieve a high generalization performance, we exploit geometric patterns that result from the fact that several crops are sown in rows. Within a field of row crops (such as sugar beets or corn), the plants share a similar lattice distance along the row, whereas weeds appear more randomly. In contrast to the visual cues, this geometric signal is much less affected by changes in visual appearance. Figure 5.8 depicts an example of an image sequence of length $S = 5$ consisting of 4-channel $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images.

In Section 4.4, we *explicitly* design a probabilistic plant arrangement model to take information about the spatial arraignment of the plants and weeds into account. In this chapter, however, we present a way for our fully convolutional neural network-based approaches to exploit the geometric signal of the plant's arrangement in an *implicit* way. Contrary to the design of handcrafted features, such as coordinate differences between plants, we let the network learn this information directly. Therefore, we propose a novel, vision-based plant classification approach that operates on image sequences.

To allow the network to access information about the plant's arrangement, we modify our presented approaches FCN and FCN-STEM by a novel architectural extension. We call this extension the sequential module. It enables the usage of image sequences to implicitly encode local field geometry. Concerning our two different classification tasks, i.e., crop-weed classification and joint crop-dicot-grass classification and stem detection, we propose to modify the respective architectures in a way that they can deal with images sequences. We propose (i) FCN-SEQ as a modification of FCN and (ii) FCN-SEQ-STEM as a modification of FCN-STEM. In both cases, the main difference is that the sequential version uses our novel sequential model, whereas the architectures for single-image processing do not.

The use of the sequential module leads to a better generalization performance of the classifier in previously unseen field environments, even if the visual appearance or the growth stage of the plants changes between training and test time. The proposed networks FCN-SEQ and FCN-SEQ-STEM are end-to-end trainable and also rely neither on pre-segmentation of the vegetation nor on any kind of handcrafted features. In the following Section 5.4.3, we describe our FCN-SEQ approach for plant classification using images sequences, and in Section 5.4.4, we describe the FCN-SEQ-STEM approach for joint stem detection and plant classification.

5.4.1 Plant Classification using Image Sequences

Figure 5.9 depicts the $S = 5$ selected $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images for building the sequence $\mathcal{I} = \{I_t, \ldots, I_{t-4}\}$ as input to our pipeline as well as exemplary predictions and their corresponding ground truth label masks. To exploit as much spatial information as possible with a small number of images, we select those images along the traversed trajectory that do not overlap in object space but have the smallest possible gap over the observed field area between each other. Figure 5.9 (left) depicts the BoniRob traversing a crop row and highlights the respective footprints of the images, which we select to build a sequence. For the image selection procedure, we use the odometry information and the known calibration parameters of the camera. The rightmost image I_t refers to the current image, whereas $I_{t-1}, \ldots, I_{t-4}$ are selected non-overlapping images from the history of acquired images.

By inspecting the image sequence shown in Figure 5.9, we can visually recognize both the crop row and also the similar spacing between the crop plants. This spacing is also called the intra-row space. Moreover, we can also observe that the weeds do not follow any specific spatial distribution and grow somewhat randomly in the field.

5.4.2 Sequential Module

In order to learn features encoding the crop-weed arrangement, the network needs to take the whole sequence that corresponds to a crop row into account. The sequential

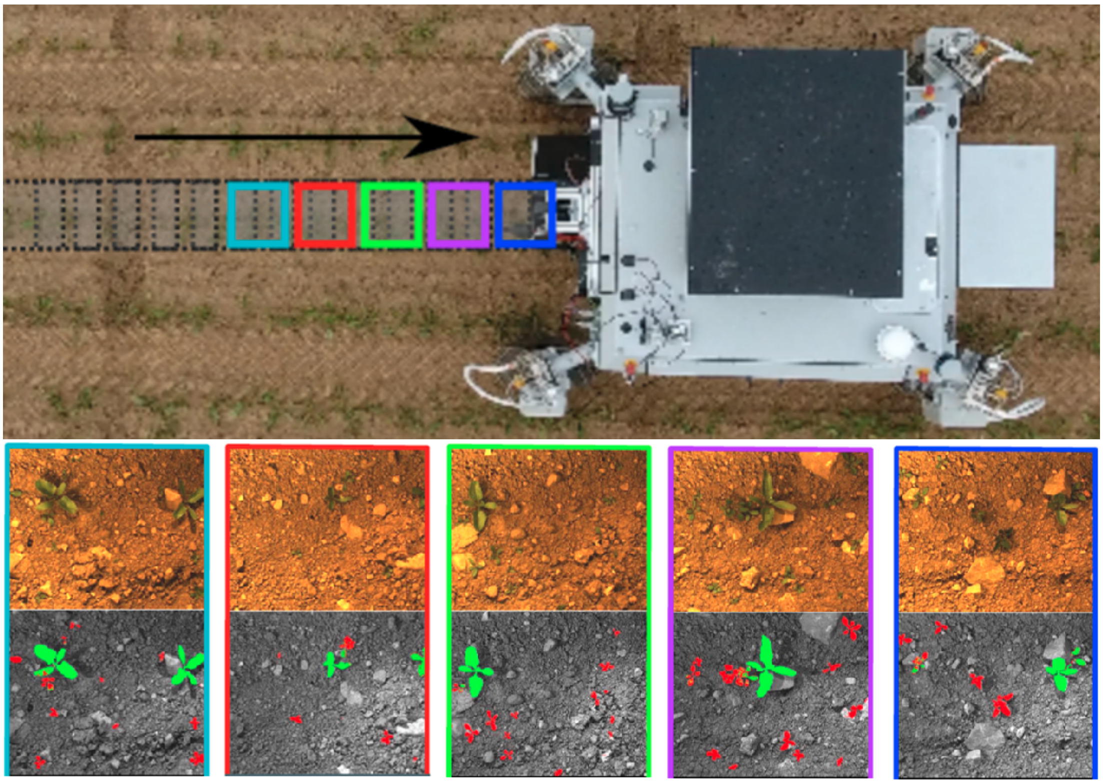


Figure 5.9: BoniRob acquiring images while driving along the crop row. Our approach exploits an image sequence by selecting those images from the history that do not overlap in object space. Exemplary prediction of crop plants and weed for the entire image sequence from the STUTT-CW-15 dataset. Note that these classification results are achieved by the FCN-SEQ approach when the model was solely trained on the BONN-CW-16 dataset.

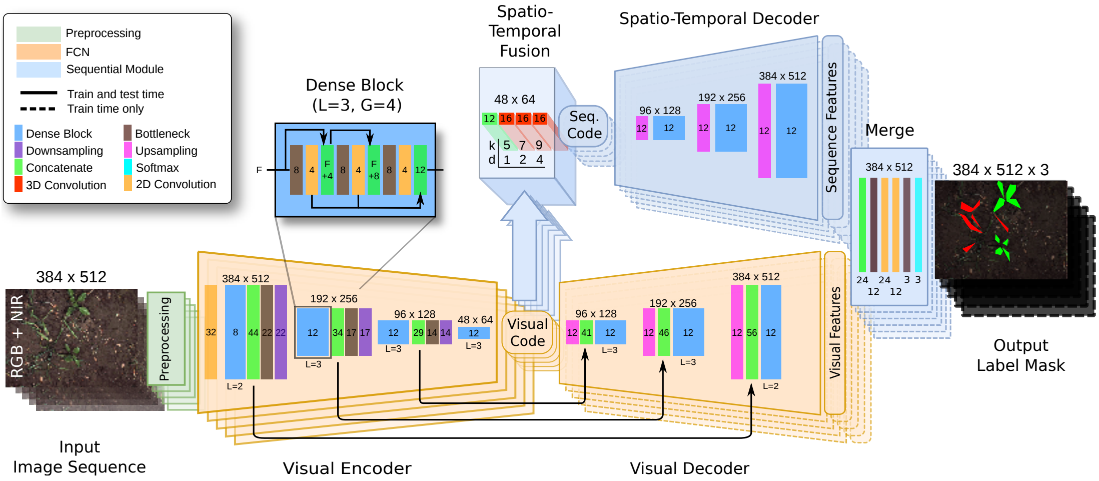


Figure 5.10: FCN-SEQ architecture. Given a sequence of S images, we first compute S visual code volumes. The visual codes are then again decoded, resulting in the visual features, and passed to the sequential module, resulting in the sequence code, which are also upsampled to full resolution by an independent spatial decoder. Finally, the visual and sequence features are merged, resulting in a pixel-wise probability distribution $\mathbf{I}_{\omega^{cws}}$. In the spatio-temporal fusion, we included the parameters k and d of the dilated convolution.

module represents our key architectural design contribution to enable sequential data processing. It can be seen as an additional parallel pathway for the information flow and consists of three subsequent parts, i.e., (i) spatio-temporal fusion, (ii) spatio-temporal decoder, and (iii) merge layer. Figure 5.10 illustrates the three parts of the sequential module and shows how they are embedded into the whole architecture.

The spatio-temporal fusion is the core part of the sequential processing. First, we create a sequential feature volume by concatenating all visual code volumes of the sequence along an additional time dimension. Second, we compute a spatio-temporal feature volume, the sequence code, as we process the built sequential feature volume by a stack of three 3D convolutional layers. Here, the sequential module aggregates the S visual codes and outputs a single sequence code, which contains information about the sequential content. We define the 3D convolutional layer analogous to the 2D convolutional layer, i.e., a composition of convolution, ReLu, batch normalization, and dropout. In each 3D convolutional layer, we use 16 3D kernels with a size of $[5 \times 5 \times S]$ to allow the network to learn weight updates considering the whole input sequence. We apply the batch normalization to all feature maps jointly regardless of their position in the sequence.

To allow the network to potentially exploit even more context information from the sequence, e.g., to extract the geometric arrangement pattern of the plants, we propose a further architectural design choice. We increase the receptive field for subsequently applied 3D convolutional layers in their spatial domain. To achieve this, we increased the kernel size k and the dilation rate d of the 3D kernels for subsequent 3D convolutional layers. We increase k and d only for the spatial domain of the convolution, i.e. $[k \times k \times S]$ with $k = \{5, 7, 9\}$ and $[d \times d \times 1]$ with $d = \{1, 2, 4\}$. This leads to a larger receptive field of the spatio-temporal fusion, allowing the model to consider the entire encoded content of all images along the sequence. In our experiments, we show that the model gains performance by using an increasing receptive field within the spatio-temporal fusion.

The spatio-temporal decoder is the second part of the sequential module that upsamples the produced sequence code to the desired output resolution, resulting in the sequence features. Analogous to the visual decoder, we recurrently perform the upsampling followed by a bottleneck layer and a dense block to generate a pixel-wise sequence features map. To achieve an independent data processing of the spatio-temporal decoder and visual decoder, we neither share weights between both pathways nor connect them via skip connections with the encoder.

The last building block of the sequential module is the merge layer. Its main objectives are to merge the visual features with the sequence features and to compute the label mask as the output of the system. First, we concatenate the input feature volumes along their feature axis and pass the result to a bottleneck layer using 12 kernels, where the actual merge takes place. Then, we pass the resulting feature volume through a stack of two 2D convolutional layers. Finally, we convolve the feature volume into the label mask using a bottleneck layer with C kernels for respective class labels and perform a pixel-wise softmax along the feature axis. Figure 5.10 illustrates the details of the specific number of layers and parameters. Furthermore, in our experiments in Section 6.3.2, we evaluate the influence of our key architectural design decisions.

5.4.3 FCN-SEQ: Architectural Concept

Our proposed FCN-SEQ approach is an extension of the FCN approach through the integration of our proposed sequential module. Thus, we transform the FCN model for single images into a sequence-to-sequence model. Analogous to the FCN, the input is either given solely by $\mathbf{I}_{RGB}$ or $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ images and the output is given by $\mathbf{I}_{\omega^{cws}}$ considering the classes $\omega^{cws} \in \{\omega_{\mathsf{c}}, \omega_{\mathsf{w}}, \omega_{s}\}$. Figure 5.10 depicts the conceptual graph and the information flow of FCN-SEQ from its input to its output for a sequence $\mathcal{I}$ of length $S = 5$. We divide the overall architecture into three main blocks: (i) the preprocessing block (green), (ii) the encoder-decoder fully convolutional neural network (orange), and (iii) the sequential module (blue).

The S visual encoders share their weights along the time axis such that we reuse it as a task-specific feature encoder for each image separately. They can also be seen as one visual encoder, which is applied S times. This leads to the computation of S visual codes being a compressed, but highly informative representation of the input images.

We route the visual code along two different paths within our network. First, each visual code is passed to the decoder of the encoder-decoder fully convolutional neural network, also sharing its weights along the time axis. Analogous to the S visual code volumes, this computation also leads to S decoded visual features volumes. Thus, the encoder-decoder fully convolutional neural network is applied to each image separately. Second, all visual code volumes of the sequence $\mathcal{I}$ are passed to the sequential module.

The sequential module processes the S visual codes jointly as a sequence by using 3D convolutions and outputs a sequence code, which contains information about the sequential content. The sequence code is then passed through a spatio-temporal decoder to upsample it to the same image resolution as the visual features, the sequence features. The resulting visual feature maps and the sequential feature maps are then merged to obtain the desired label mask output.

To facilitate the recovery of spatial information, we use skip-connections and concatenate feature maps produced by the dense blocks in the encoder with the corresponding feature maps produced by the upsampling in the decoder and feed both feature volumes into a bottleneck layer to fuse them. In contrast to the encoder, we reduce the systematic increase of feature maps for the decoder by omitting the concatenation of the input of a dense block with its respective output.

We designed the FCN-SEQ approach to have different behaviors during training and test time. During training time, we treat it as a sequence-to-sequence model. Thus, we use all S predictions to obtain the loss signal, which is responsible for the weight updates during the backpropagation step of the training. During test time, however, we treat the network as a sequence-to-one model, only computing the visual features in the decoder part for the latest acquired image in the sequence.

5.4.4 FCN-SEQ-STEM: Architectural Concept

This section describes our last fully convolutional neural network-based approach. FCN-SEQ-STEM is an extension of the FCN-STEM approach through the integration of our proposed sequential module. It performs joint stem detection and pixel-wise classification of plants by analyzing image sequences. Analogous to FCN-STEM, the input is either given solely by $\mathbf{I}_{RGB}$ or $\mathbf{I}_{RGB}$+$\mathbf{I}_{NIR}$ and the output is given by $\mathbf{I}_{\omega^{cdgs}}$ considering the classes $\omega^{cdgs} \in \{\omega_\mathsf{c}, \omega_d, \omega_g, \omega_s\}$.

In the following discussion, we will not distinguish between the plant and stem decoder, since they are architecturally the same. However, note that the decoders for stems and plant segmentation do not share weights. We will use the term visual decoder to denote both decoders.

Except for the sequential architectural parts, the FCN-SEQ-STEM network follows the structure of its non-sequential counterpart FCN-STEM. For the sequential module and the spatio-temporal decoder, however, FCN-SEQ-STEM follows the FCN-SEQ architecture.

Figure 5.11 shows the network architecture and the information flow from the input to the output for a sequence $\mathcal{I}$ of length $S = 5$. Conceptually, we have a network with one visual encoder and two task-specific visual decoders such as in FCN-STEM that takes only a single image as input and produces a single plant label mask and a single stem region mask. Thus, the visual encoder as well the task-specific decoders share their weights over different timesteps, respectively. This is key for using the same architecture for learning plant classification and stem locations. To integrate sequence information, we use the sequential module described in Section 5.4.2 that takes S

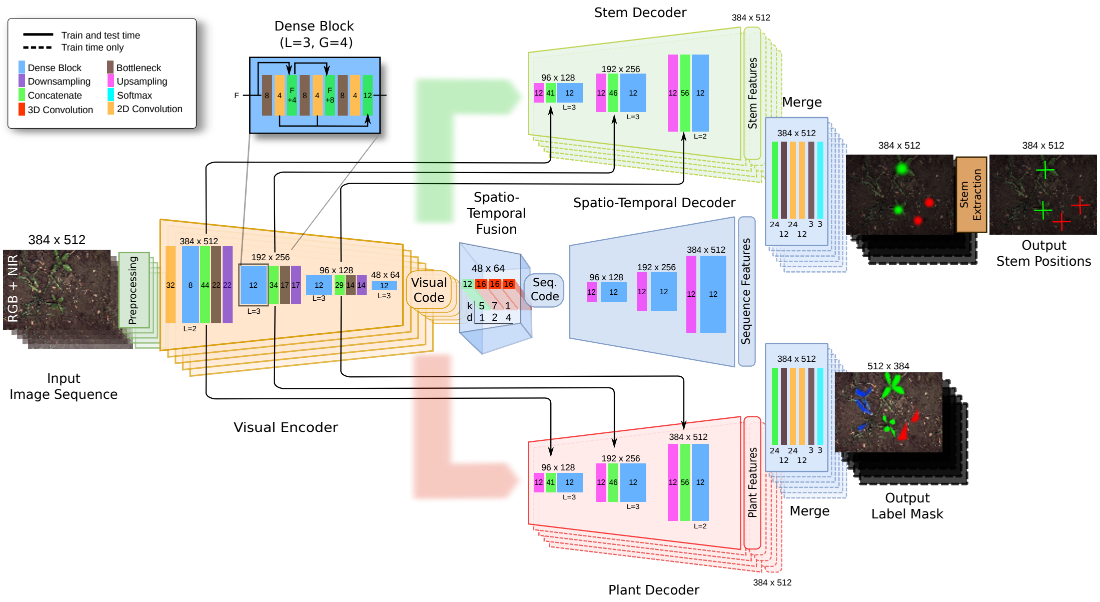


Figure 5.11: FCN-SEQ-STEM architecture for sequential joint stem and plant classification. We first encode the input images using the encoder and then pass the feature volumes to the task-specific decoders as well as to the sequential module. The respective decoded features for plant classification and stem detection are then merged with the spatio-temporal features leading to two outputs, the plant mask $\mathbf{I}_{\omega^{cdgs}}$ considering the classes $\omega^{cdgs} \in \{\omega_{\mathsf{c}}, \omega_d, \omega_g, \omega_s\}$ for the pixel-wise classification of the plants and the stem mask $\mathbf{I}_{\omega^{cds}}$ considering the classes $\omega^{cds} \in \{\omega_{\mathsf{c}}, \omega_d, \omega_s\}$ for the segmentation crop-weed stem regions. Finally, we extract the stem positions from the stem mask in the stem extraction.

encoder feature volumes and produces the sequence features. The sequence features are then merged with the outputs of the S task-specific decoders for plant classification and stem detection. Finally, we extract pixel-accurate stem positions from the stem mask using the postprocessing step as described in Section 5.3.2.

5.5 Summary

In this chapter, we presented different approaches to the fully convolutional neural network-based plant classification. All these approaches follow the main pipeline shown in Figure 5.1.

First, we propose our basic architectural design, i.e., our FCN approach, which is a self-designed, lightweight encoder-decoder structured network architecture that performs a pixel-wise classification.

Subsequently, we extend our basic FCN approach to also processes UAV images. Therefore, we adjust the receptive field to use a larger area in the image for the classification of a pixel. We call this approach FCN-UAV. This approach can implicitly incorporate information about the relative arrangement of plants in the feature extraction as in considers a larger part of the image for optimizing its parameters.

We introduce our FCN-STEM approach, which represents a single model for jointly determining stem locations of plants and the pixel-wise classification of plants. In addition, we present FCN-UAV-STEM for the application of plant counting.

Next, we propose to exploit additional geometric information about the local arrangement of the plants by analyzing image sequences that implicitly carry the information about the plant arrangement for a local field strip. Therefore, we design the sequential module, which is a subnetwork that analyzes visual features of consecutive images from a sequence and extracts spatio-temporal features that encode the field geometry. The integration of the sequential module into our approaches FCN and FCN-STEM leads to our contributions and novel approaches FCN-SEQ and FCN-SEQ-STEM, respectively.

All UGV approaches achieve a suitable runtime for online in-field operations such as selective spraying or mechanical weed control.

Chapter 6

Experimental Evaluation

In Chapter 4 and Chapter 5, we present different vision-based plant classification approaches based on either traditional machine-learning techniques such as random forests or modern machine-learning techniques such as fully convolutional networks. All systems aim at analyzing images to predict the type, location, and spatial extent of the crop plants and weeds at pixel-level. On the one hand, the classification systems provide mobile robots with the classification results online, so that the robot can carry out actions, such as selective weed control, while driving over the field. On the other hand, the UAV-based classification provides crucial information for crop monitoring applications.

In this chapter, we present a comprehensive evaluation of our proposed classification systems. We collected an extensive database between 2015 and 2019, consisting of around 26,500 manually labeled images acquired by different UGVs and UAVs in various field environments located across central Europe. Labeling these 26,500 images was a task of approximately six person-months and was a substantial effort to realize the evaluation described in this chapter.

The crop considered in this work are sugar beets, an important row crop in Germany, and other countries of Northern Europe. This database allows us to assess the performance under challenging real-world conditions and to focus on different aspects.

For UGVs, we assess the performance of all our classification systems in terms of their ability (i) to separate the vegetation from the background, (ii) to classify crop plants and weeds enabling robots for selective treatments, and (ii) to classify crop plants, dicotyl weeds, and grass weeds along with detecting the stem location of the plants and weeds. The latter classifier enables robots for selective and plant-specific high precision treatments. In the case of the UAVs, we assess the performance under the aspects of (i) monitoring the spatial distribution of crop plant and weeds, but also in terms of multiple weed species and (ii) plant counting.

6.1 Evaluation Objectives

We design the experiments in this chapter to explicitly analyze the classification performance under the following aspects: for (i) classification performance, (ii) robustness to changing field conditions, (iii) effectiveness to adapt to new field conditions, and (iv) runtime.

Classification performance We pursue two different ways to investigate the performance of a classifier. First, we report the achieved performance in a pixel-wise manner. This measure corresponds to the coverage of the crop plants and weeds in the scene. For this metric, we compare every pixel of the prediction with the associated pixel of the ground truth. Thus, the resulting precision and recall measures reflect semantic segmentation performance. The advantage of this metric is that the performance can be read directly in terms of coverage by individual plants or weeds in the image space. The downside of this metric is that the performance values often turn out to be to the disadvantage of the classes with a lower probability of occurrence.

Second, we evaluate the classification systems in terms of an object-wise performance, i.e., a metric for measuring the performance closer to the plant-level. Here, we compare the predicted label mask with the class-wise connected components, e.g. crop plants and weeds, from the corresponding ground truth. Compared to the pixel-wise metric, where we perform precision, recall, and F1-score based on the comparison of individual pixels, the object-wise metric is based on the comparison of objects, i.e., connected components. The pixel-wise overlap controls the data association between objects in the prediction and the ground truth. Two objects are considered to correspond to each other if their overlap is at least of an IoU ≥ 0.5. Throughout the thesis, we only consider plant segments with a minimum size of $0.15\,\mathrm{cm}^2$ in object space. We consider smaller objects to be noise as they are only represented by a few pixels and would therefore bias the performance.

For both cases, we report the class-wise F1-score (F1), recall and precision in percent as well as the average across all classes. We choose these metrics for our evaluation, as these are interpretable values and allow a direct assessment of the expected performance in the real-world. For the stem detection task, we analyze the mean average distance (MAD) representing the error between detected to the actual stem locations. We provide details on these metrics in Section 2.4.

Robustness to changing field conditions A focus of this thesis is the development of crop-weed classifiers that aim at bridging the lack of classification performance in new fields. Therefore, we explicitly evaluate the generalization capabilities of the classification systems to new and changing field conditions. We explicitly analyze the classification performance on datasets that were acquired in different field environments. Therefore, we train a classifier on data from a particular field and deploy it on data acquired in another field. We explicitly choose our datasets such that they contain various growth stages of crop plants and weeds, different weed species, and different soil conditions. Throughout this chapter, we refer to these experiments as “evaluation under changing conditions”.

Effectiveness to adapt to new field conditions The transferability of a classifier means that its statistical model can be adapted to perform appropriately on new data that was not part of the initial training phase. We measure the transferability through the required effort to adapt the classification model to a new dataset. Precisely, we measure this effort in the number of extra labels that have to be created to re-train the classifier for the target data in order to achieve a particular performance.

Runtime A model suited for online plant classification on a mobile robot must work with a minimum required runtime in order to achieve a sufficient throughput at particular driving speed. The maximum speed of the BoniRob for the datasets, which we present in Chapter 3, was $0.3\,\frac{m}{s}$. In this thesis, however, we assume a maximum driving speed of about $1.4\,\frac{m}{s}$, i.e., of $5\,\frac{km}{h}$. Regarding our acquisition setup for UGVs, which we describe in Section 3.1.1, an image covers about 300 mm of the object space along the driving direction. Thus, the algorithms have to analyze the image with at least 4.7 Hz. Otherwise, the area under the robot cannot be analyzed entirely during travel. Note that this constraint does not hold for the development of the UAV-specific classification models. For the UGVs, we evaluate the runtime of our approaches in Section 6.11.

A further development goal of this thesis is to define a single classifier design, which is suitable to provide high performance and generalization capabilities for the plant classification tasks. Given a single and consistent architectural design, we can re-train or adapt the same network as new data arrives through re-using pre-trained weights, and we avoid to perform expensive hyperparameter searches for different architectures.

Table 6.1 summarizes our proposed approaches that we evaluate in the context of this experimental evaluation. Note that we have already shown Table 6.1 in Chapter 1. For better readability and handling of the document, we show the overview of the approaches here again. The table provides for each approach its name, an abbreviation, the application, and a small description. In the first order, we distinguish between random forests and fully convolutional networks. Additionally, we divide the experiments into three categories. The first category focuses on experiments with UGVs, which perform the classification solely based on visual information, i.e., exploiting visual features extracted from the images. The second category covers all UGV-based approaches that take into account additional geometric information about the spatial distribution of plants and weeds. The third category covers our approaches optimized for use on UAV images. Across all experiments in this chapter, we compare the performance of the random forest with that of fully convolutional neural networks. What are the advantages and disadvantages of the two different paradigms? Is one of the two methods generally more suitable? We contribute to these questions within our experiments. In the case of the UAV experiments, we solely consider RGB data as input to the classifiers. For the UGVs, however, we have access to additional NIR information. Many of the experiments presented here investigate the use of NIR information

Table 6.1: Overview of our proposed random forest-based and fully convolutional neural network plant classification systems, which we evaluate in this chapter. Each approach has been either entirely or partially presented in our published conference papers [82, 85, 86, 88, 89] or journal articles [83, 84, 87].

Description	Random Forest	FCN	Description
Visual Plant Classification Described in sections 4.3 and 5.2			
Keypoint-based approach classifying lattice-spaced keypoints	RF-KP [81, 86]	FCN [82]	Fully convolutional neural network for plant classification on single images
Object-based approach classifying connected vegetation components	RF-OBJ [87]		
Cascaded approach combining RF-KP and RF-OBJ	RF-CAS [87]		
		FCN-STEM [82]	FCN for plant classification and stem detection on single images
Visual and Geometrical Plant Classification Described in sections 4.4 and 5.4			
Geometric classifier exploiting plant arrangement	GC [89]	FCN-SEQ [84]	Sequential FCN for plant classification on image sequences
Semi-supervised approach exploiting visual RF-CAS and and geometric GC classifier	RF-GC [89]		
		FCN-SEQ-STEM [83]	Sequential FCN for plant classification and stem detection on image sequences
UAV-Based Plant Classification Described in sections 4.5 and 5.2.2			
RF-CAS exploiting geometric features for UAV imagery	RF-UAV [88]	FCN-UAV [85]	FCN for plant classification and stem detection on UAV images exploiting larger spatial context
		FCN-UAV-STEM	FCN-STEM applied crop counting and plant classification based on UAV imagery

for plant classification. The camera systems currently available on the market, which provide RGB and NIR image data, are typically more expensive. Therefore, we analyze the necessity of NIR information for the plant classification task.

6.2 Experimental Outline and Summary of the Results

In this section, we outline and summarize all the experiments we conduct in this chapter and summarize the main conclusions. Due to a large number of experiments, we provide an overview here instead of a discussion at the end of this chapter. We refer to the individual experiments in the respective section, which deals with them in detail. For the readers with limited interest in detailed experiments, skip those and proceed to the section of particular interest.

Classification Models In Section 6.3, we evaluate the architectural design choices for random forests and fully convolutional neural network classifiers. For both types of classification models, we evaluate a common set of hyperparameters that we use for the experiments throughout this chapter. In certain experiments, however, the hyperparameters may deviate from the basic set in order to investigate specific aspects of the approaches. In this case, we explicitly point this out.

We motivate the use of our RF-CAS approach for the plant classification using random forests. RF-CAS is a classification model for the visual plant classification and is a cascade of the two subordinated keypoint-based RF-KP and object-based RF-OBJ approaches. RF-KP can deal with overlapping plants and weeds in the images but at the cost of being computationally expensive. RF-OBJ has the advantage that it is substantially faster but cannot deal with overlapping crop plants and weeds. We show in our experiment in Section 6.3.1.2 that RF-CAS can exploit the respective advantages of these approaches while compensating their drawbacks, i.e., being fast and able to deal with overlapping plants and weeds. Finally, we provide an analysis of the importance of the keypoint-based and object-based features in Section 6.3.1.3. We show that NIR information plays an essential role in our proposed random forest-based plant classification. Regarding the fully convolutional neural networks, we show that our lightweight, self-designed model architecture outperforms state-of-the-art network architecture for pixel-wise plant classification.

Vegetation Classification In Section 6.4, we analyze the performance of the threshold-based vegetation classification step that is an essential part for any random forest-based approach in this thesis. Furthermore, we compare the performance the threshold-based approach with the one obtained by our FCN approach.

First, we investigate different vegetation indexes in Section 6.4.1 for the threshold-based vegetation classification and find that the best performing vegetation index is the NDVI, if RGB+NIR is available, and the ExG if only RGB data is available. We show that the NIR information is especially beneficial for the performance under changing field conditions. Furthermore, we show that automated detection of a threshold by Otsu's method [104] fails in situations with imbalanced class occurrences and, thus, is not suitable for the vegetation classification task. Thus, to achieve a suitable performance, the threshold has to be selected manually. This reflects a limitation for the autonomous deployment, as human intervention is needed.

Second, we show in Section 6.4.2 that our FCN approach provides a high vegetation classification performance when exploiting RGB+NIR or solely RGB data. The FCN performance is stable under similar as well as under changing field conditions. We conclude that fully convolutional neural networks are well suited for the task and recommend using them in cases when solely having access to RGB data.

The only case in which we see threshold-based approaches as advantageous is when on the one hand, NIR information is available, and on the other hand, a human operator supervises the system and can adapt the threshold to changing situations. In such a situation, the labeling of data and training of classifiers can be circumvented.

Vision-Based Crop-Weed Classification In Section 6.5, we perform experiments analyzing the vision-based classification performance of our random-forest and fully convolutional neural network approaches, i.e., RF-CAS, FCN, and FCN-RGB, where FCN-RGB refers to the RGB-only variant of FCN. We demonstrate that the fully convolutional neural network-based approaches provide better performance compared to the random forest-based approach in both cases, under similar and changing field conditions. We argue that the learned features of the fully convolutional neural network approaches are more descriptive for the task, provide a better capacity, and generalize better in new fields compared to the handcrafted features used in the random forest.

We show in Section 6.5.2 that the crop-weed classification using fully convolutional neural networks is comparable when exploiting solely RGB or RGB+NIR information under similar field conditions. Thus, having access to labels of similar field conditions can compensate for the use of additional NIR information. Under changing field conditions, however, neither the fully convolutional neural network nor the random forest-based approaches provide suitable performance to be capable of being applied in real-world applications when the classifier is trained once and then deployed in new and unseen field environments (Section 6.5.1). Adding to this, we also show that a greater diversity of the training data, e.g., data from different field environments, helps to learn better features to generalize to new field environments.

Vision-Based Crop-Weed Classification Exploiting Plant Arrangement In Section 6.6, we evaluate our approaches RF-GC and FCN-SEQ that additionally exploit the spatial arrangement of the crop plants and weeds within local field strips. Both classification systems use sequences of images as input and combine visual features along with additional geometric features encoding the spatial patterns of plant locations resulting from the sowing process. The goal of our sequential approaches is the development of crop-weed classifiers that bridge the lack of generalization capabilities to new field environments. In these experiments, we evaluate the sequential classification approaches under similar and changing field conditions and compare the achieved performance with the one obtained by the non-sequential approaches. Here, we demonstrate our experiments demonstrate superior generalization capabilities for the fully convolutional neural network approaches exploiting the plant arrangement signal and show that the spatio-temporal features are key supporters for the performance under changing field conditions. Nevertheless, exploiting the geometric features also helps to better performance under similar field conditions.

For our random forest-based visual and geometric classification system RF-GC, however, we observe that it cannot reliably adapt to changing field conditions if no training data is available from the targeted field environment to properly initialize its visual and geometric classifier. Under similar field conditions, the system can be adequately initialized and can compensate for the limited capacity of the handcrafted visual features to some degree. In Section 6.6.4, we furthermore demonstrate the ability of FCN-SEQ and RF-GC to extract features encoding the relative arrangement of the plants through an experiment with simulated data. In this experiment, we switch off visual information about the color and shape of the plants and show that our sequential approaches are capable of detecting the plants and weeds only based on geometric information about their spatial distribution. Finally, we evaluate the effect on the performance when neglecting the NIR information. We can show that using additional NIR information aids the generalization capabilities of FCN-SEQ compared to its RGB variant FCN-SEQ-RGB to new and changing field conditions.

Joint Plant and Stem Detection for Species-Specific Treatments In Section 6.7, we conduct experiments to analyze the quality of our vision-based classification pipelines for joint pixel-wise plant classification and stem detection enabling selective and plant-specific treatments. For these experiments, we rely on a different database compared to the previous experiments, as we need labeled data that additionally considers the locations of plant and weed stems as well as an additional grass weed class for the plant classification. For the additional task of stem detection, we extend our fully convolutional network architectures with an additional task-specific decoder.

First, we compare our proposed FCN-SEQ-STEM approach against its non-sequential version FCN-STEM. We show that approaches for joint plant classification and stem detection provide suitable performance for high precision plant-specific treatments under similar field conditions. The stem detection works properly and provides the stem locations within a spatial precision of around 2-4 mm.

Under changing field conditions, we can show that the fully convolutional neural network approaches exploiting the plant arrangement signal support superior generalization capabilities. Furthermore, we analyze the effect of using two task-specific decoders for plant classification and stem detection within one the use task-specific decoders, which share a single encoder for the feature extraction, aid the performance for the plant classification and the stem detection.

Supervised Classifier Transfer in the Context of Labeling Effort In Section 6.8, we evaluate the effectiveness of our classifiers to adapt to new and changing field conditions. We investigate the performance of the classifiers FCN, FCN-SEQ, and RF-GC in new and changing field environments. We analyze how efficiently we can adapt the classifiers to the environmental conditions of the targeted field by re-training them with only a small amount of labeled data. We demonstrate that our proposed RF-GC approach provides the best adaptability, as it adequately performs in new field environments requiring only a labeling effort of one minute by a human operator.

We furthermore show that the sequential FCN-SEQ approach can better exploit smaller amounts of data to adapt to new and changing field conditions compared to its non-sequential variant FCN.

UAV-Based Plant Classification for Automated Crop Monitoring In Section 6.9, we analyze the performance of our UAV-based plant classification systems for the application of crop-weed classification, multi-species classification, and plant counting. For the crop-weed classification task, we analyze the performance for high- and low-resolution imagery, respectively. Note that for all UAV experiments, we solely consider RGB images as the input to the classification systems. First, we show that for high-resolution UAV images, our fully convolutional neural network provides better crop-weed classification performance under similar as well as under changing field conditions than our random forest-based approach. A major reason for the better performance is that fully convolutional neural networks can adequately deal with the separation of vegetation and soil using RGB data. These results are in line with the vegetation classification results in Section 6.4. Besides also for the multi-species classification, the convolutional networks perform better compared to the random forest.

Second, the same observation holds also for low-resolution UAV images. Here, our FCN-UAV approach performs better than the RF-UAV approach. Especially in cases where plants overlap, the fully convolutional network approach can show its advantages. However, the explicit modeling of geometric features in the random forest-based RF-UAV approach helps to bridge the performance loss when deploying the classifiers in new and unseen field environments.

Third, we demonstrate that our FCN-UAV-STEM approach, i.e., the approach, we originally developed for joint plant classification and stem detection, is suitable to perform UAV-based automated crop counting. Even under harsh field conditions concerning high weed pressure and overlapping crop plants, our approach provides results that are better compared to human performance.

Runtime Performance for In-Field Treatments In Section 6.11, we analyze the runtime of our UGV-based plant classification approaches. We show that all proposed approaches for this purpose obtain a fast enough runtime of >5 Hz enabling agricultural robots to perform on-field weed control. Our fastest approach is FCN. It can infer up to 33 images per second.

6.2.1 General Experimental Setup

In this section, we describe general conditions that apply to all experiments in this chapter. If an experimental design deviates from these general conditions, we explicitly state this in the respective section.

In the performance tables, we report the class-wise and average F1-score (F1), precision (P) and recall (R) in percent to analyze the pixel-wise classification performance and also the object-wise performance. The term “training” refers to the training data for a particular experiment, whereas the term “deployment” refers to the used test dataset on which we report the performance. We report the F1-score for the soil class as the values for the corresponding precision and recall mostly range between 99.5 % and 99.9 %. Note that the performance tables always report the result that is achieved under a class labeling according to Equation (2.23). For most of the experiments, we additionally provide precision-recall curves evaluating the class-wise performance of individual approaches in more detail and to compare different classifiers. For any object-wise evaluation in this chapter, we consider objects which are of a size of around $\geq 0.15\,\text{cm}^2$, as smaller objects are represented only by too few pixels in the image and, thus, considered to be noise. Note that the evaluation in our related conference papers [82, 85, 86, 88, 89] and journal articles [83, 84, 87] mostly considers a evaluation of objects at a size of $\geq 0.5\,\text{cm}^2$. The evaluation in this thesis, however, is somewhat more challenging, as typically smaller objects are harder to predict well. We unified all boundary conditions for the evaluations provided in this thesis to ensure fair comparisons.

For a fair comparison between the random forest-based and fully convolutional neural network approaches, we train all classifiers from scratch. This means that we do not use pre-trained weights for the initialization of the networks and that we do not adapt a threshold for the vegetation classification module of the random-forest-based during the deployment phase. A challenge for the training of neural networks lies in the choice of the number of training epochs. Too many epochs for training can lead to model overfitting on the training data set — too few epochs for training lead instead to an underfitting of the model. Early stopping is a data-driven method that allows the training process to be based on the development of performance on a validation data record not used in training. In general, the goal of early stopping is to stop training as soon as the model performance stops improving on the validation data set. In the experiments in this chapter, we stop the training of the fully convolutional neural networks when the average F1-score for the pixel-wise classification performance on the validation data does not increase at least 1 % over five epochs of training. In the case of the random forests, none of the approaches in these experiments use the validation data split.

Throughout our experiments, we analyze the performance of our proposed approaches under two different aspects. First, we test the performance under similar field conditions. This means that the classifier is trained on data coming from one or more field environments and is then deployed on a held-out portion of the test data coming from the same field(s). Through this, we measure the performance when the classifier is *aware* of the present field conditions in the data, as it was on data that comes from the same field. Note that we never report the performance on images that are part of the training data itself. Second, we test the performance under *changing field conditions*. We define changing field conditions if the training data and test data are acquired in different field environments and may be captured by different robots. Thus, in these experiments, a classifier is *not aware* of the field conditions within the deployment phase on the test datasets. The performance under changing field conditions reflects the generalization capabilities of the classification system to new and changing field conditions.

For the UGV experiments, we evaluate under similar and changing field conditions in the following scheme. For the case of similar field conditions, we analyze the performance on the respective largest dataset, i.e., on BONN-CW-16 for the crop-weed classification and BONN-CDGS-16 for the crop-dicot-grass and stem classification. Additionally, we combine all available data sets into one overall data set, i.e.,

$$\begin{aligned}\text{ALL-DATA-CW} = \{&\text{BONN-CW-16},\\ &\text{BONN-CW-17},\\ &\text{STUTT-CW-15},\\ &\text{ANCONA-CW-18},\\ &\text{ZURICH-CW-16}\},\end{aligned}$$

$$\begin{aligned}\text{ALL-DATA-CDGS} = \{&\text{BONN-CDGS-16},\\ &\text{STUTT-CDGS-15},\\ &\text{ANCONA-CDGS-18},\\ &\text{ZURICH-CDGS-17}\}.\end{aligned}$$

6.2.2 Training, Validation, and Test Data Splits for UGV Datasets

The structure of the expected input data differs according to the respective classification models that we deploy on the UGVs. We distinguish between approaches that operate on, single images, i.e., RF-OBJ, RF-KP, RF-CAS, FCN, FCN-STEM, and sequential approaches that operate on image sequences, i.e., FCN-SEQ and RF-GC. To ensure a fair comparison between all approaches, we use the same image data for training, validation, and testing, respectively. We split each UGV dataset into chunks of fixed size. This initial step is required as the sequential approaches require consecutively acquired images as their input.

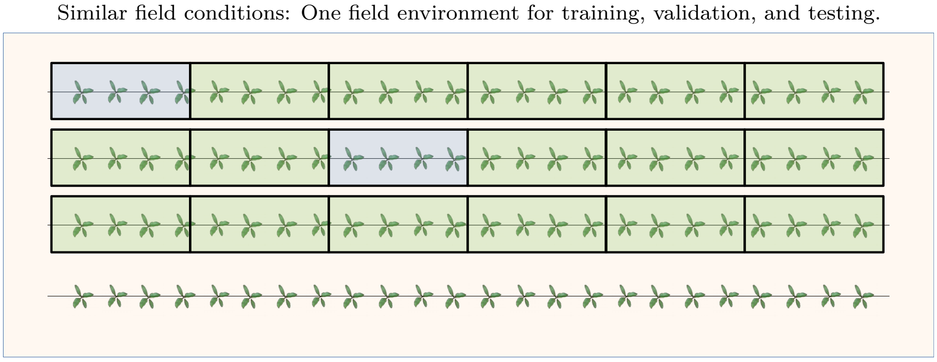

Similar field conditions: One field environment for training, validation, and testing.

Figure 6.1: Data split policy for the experimental evaluation under similar field conditions. We split the data acquired in a particular field environment into chunks for the training (green) and validation (blue), testing (no color)

Figure 6.1 illustrates how we select the training, validation, and test split for the evaluation under similar conditions. Here, we use data coming from one field environment. We divide the dataset into chunks of consecutively acquired images. The length of the chunks depends on the total size of the data set but is always a minimum size of 6 m along the row in object space. The width of a chunk is defined by the field of view of the camera system, i.e., around 50 cm. For an experiment under similar conditions, we split off the test portion of the data consisting of chunks (no color) that are located in are a particular region of the field. By this, we ensure that we do not mix the test split with data for the training and validation splits. We then separate the remaining chunks into the training data (green) and the validation data (blue), see Figure 6.1, whereas we take care that the validation data contains an example of all considered class labels. For instance, we avoid to randomly pick a chunk for validation which does not contain weeds.

Figure 6.2 illustrates the data split for an experiment under changing conditions. Here, we use data from a particular field environment for training and validation. For testing, we use the entire data from another field environment.

For the sequential fully convolutional neural network approaches, i.e., FCN-SEQ and FCN-SEQ-STEM, we create as many sequences as possible from the chunks to obtain the respective sets of training and validation image sequences. Consider a chunk contains I consecutive images. Then, we extract $N < I$ sequences of the sequence length of S that exploit as much spatial information as possible with S images. For a single sequence, we select those images along the traversed trajectory that do not overlap in object space but have the smallest possible gap over the observed field area between each other. An example sequence of $S = 5$ images is shown in Figure 5.8. Note that images in the chunk can belong to more than one sequence. However, splitting the data on the level of chunks ensures that images from the training dataset are not part of sequences of the validation or test dataset. During training, we then randomly feed those image sequences into the network, not considering to which training-chunk they belong. By this procedure, we randomize the training as much as possible. This pushes

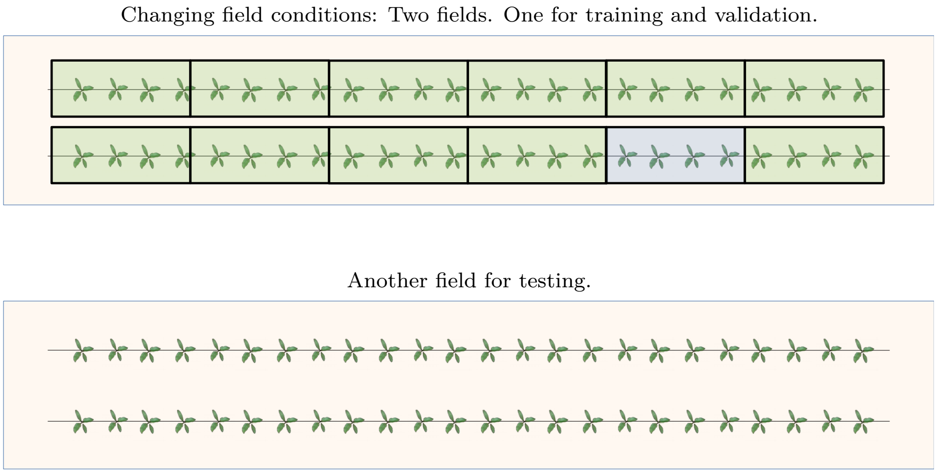

Changing field conditions: Two fields. One for training and validation.
Another field for testing.

Figure 6.2: Data split policy for the experimental evaluation under changing field conditions. Top: we split the training data (green) and validation data (blue) into chunks. For testing, we feed the entire data of the targeted field environment into the classifiers.

the network to learn more general visual as well as spatio-temporal features.

For the random forest-based approach RF-CAS, however, we feed the images into the classifier in a continuous data stream. The reason is that this approach continuously processes the subsequent images of the last two meters along the crop row to update its probabilistic arrangement model according to Equation (4.19). The data stream is only interrupted when the crop row ends or no crops are detected over a certain distance. In this case, the model is not updated again until two meters have been traveled along a new crop row. We use this policy on splitting the data throughout the entire chapter for all experiments with UGVs.

Finally, for the non-sequential approaches, we do not consider the chunk structure at all. Here, we randomly sample images from the training data and feed them into the classifier. This includes all the UAV experiments in this chapter.

6.2.3 Image Size

For all UGV experiments in this chapter, we downscale the RGB and NIR images to a width $W = 512$ and height $H = 384$. Concerning the original image size from the camera's sensor of $W = 1296$ and $H = 966$, this reflects a downscaling factor of about 2.5. The downscaling of the images leads to a resulting ground sampling distance of around 1 mm in object space. The resizing of the images has two reasons. First, this procedure reduces the required GPU memory for the fully convolutional neural networks. Due to the larger feature maps and the multiple uses of the encoder in the case of image sequences (FCN-SEQ and FCN-SEQ-STEM), the memory requirements of the models increase accordingly. Second, in the case of random forest, downscaling mainly results in a faster runtime, as fewer pixels are used for feature extraction. We downscale the RGB and NIR images before any performed processing and use a bicubic interpolation for the scaling.

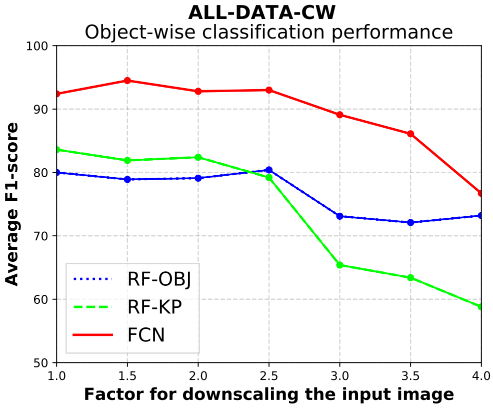


Figure 6.3: Object-wise crop-weed classification performance for image data of different size.

For both types of classification systems, either fully convolutional networks or random forests, we evaluate the effect of the downscaling on the performance for the crop-weed classification. Therefore, we train the RF-OBJ, RF-KP, and FCN approach on 70 % of the ALL-DATA-CW dataset and evaluate the performance on a 25 % held-out test portion. We repeat this experiment for no downscaling and a downscaling with factors $s \in \{1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$.

Figure 6.3 summarize the obtained object-wise performance in terms of the achieved average F1-score across all class labels for the ALL-DATA-CW dataset. We choose the object-wise metric for this evaluation, as it is more sensitive to smaller plants and less affected by changes in the plant coverage due to the downscaling of the images. The results suggest downscaling the input with up to a factor of 2.5 does not notably affect the performance of any approach. A further downscaling, i.e., a ground sampling distance of less than $1\frac{mm}{px}$, leads to a decrease in performance, as can be seen in Figure 6.3. It affects the keypoint-based approach the most. The results even indicate that larger image size can also lead to worse performance of the FCN approach. We conclude that an image size of $W = 512$ and $H = 384$ is an appropriate choice, as it reflects the best trade-off between classification performance and model size and, thus, runtime.

6.3 Plant Classification Models

The experiments in this section are designed to evaluate our architectural design decisions for our random forest and fully convolutional neural network classifiers. Furthermore, we evaluate a basic set of hyperparameters which we then use in the experiments presented in this chapter.

The main objective of the model selection and tuning of hyperparameters is to maximize the plant classification performance. The goal is to deploy classifiers, which provide a high recall and precision for the classification of crop plants and weeds, or even multiple species in the field. To a certain extent, however, this contradicts the restriction regarding the required runtime. Typically, larger models achieve better performance but provide slower runtime as more computation has to be executed.

We simultaneously address both runtime and performance of the models within the following experiments. The output of this evaluation is two basic classification models along with a set of hyperparameters that we use in the remainder of this chapter. In Section 6.3.1, we propose to aggregate our keypoint-based approach RF-KP and our object-based one RF-OBJ in a cascade resulting in our base classifier RF-CAS. Furthermore, we search for the best performing RF-CAS model by evaluating its performance under the variation of certain hyperparameters. In Section 6.3.2, we compare our proposed FCN architecture with other state-of-the-art architectures and show that our proposed model provides the best overall performance. In addition to that, we evaluate its performance to select an appropriate set of hyperparameters.

6.3.1 Random Forest Model

We design the experiments in this section to evaluate a set of hyperparameters for our proposed random forest-based classifiers and to demonstrate that our RF-CAS approach, which combines the advantages of RF-KP and RF-OBJ, provides better results in terms of overall performance and runtime.

6.3.1.1 Hyperparameter Search for the RF-CAS Model

The random forest is an ensemble of decision trees. Regarding [15], the only crucial decision-tree-specific hyperparameter to tune is the number of randomly selected features that are considered to find the best data split given the minimum Gini-score according to Equation (2.5). We additionally evaluate the number of trees forming the forest and the maximum allowed depth for a single tree.

Another important hyperparameter of the RF-CAS model, which implies the RF-KP model, is the definition of the keypoint size and the lattice distance, which controls the spacing of the keypoints in image-space.

For the experiments in this section, we use the average F1-score as the criterion for evaluating the performance of a particular hyperparameter. We solely consider the classes crop and weed. We neglect considering the soil class for the hyperparameter search as this performance is solely affected by the threshold-based vegetation classification described in Section 4.3.1.

We use the ALL-DATA-CW and ALL-DATA-UAV-CW datasets for our evaluation. Since these datasets are composed of all available crop-weed data sets, they represent challenging conditions for a classification model and are therefore ideally suited for the search for hyperparameters. First, we split the respective datasets into distinct chunks of 75 % and 25 %. We keep the 25 % split out of the experiments in this section, as we use it later as test data for the evaluation of our approaches. We divide the remaining 75 % into two equal parts and use one to train the classifiers and the other to validate the performance. We performed several experiments varying the following hyperparameters, i.e.,

$$
\begin{aligned}
\text{Number of trees } &\in \{10, 25, 50, 100, 150, 250, 500\} \\
\text{Random features } &\in \{5, 10, 15, 25, 50, 100\} \\
\text{Depth of trees } &\in \{5, 10, 15, 20, 25, 30, 40, 50\} \\
\text{Keypoint size } &\in \{[5 \times 5], [10 \times 10], [20 \times 20], [40 \times 40], [80 \times 80]\} \\
\text{Lattice distance } &\in \{3, 5, 10, 15, 20, 30, 40, 50\}
\end{aligned}
$$

We analyze the performance of the RF-OBJ and RF-KP approach for UGV images on the ALL-DATA-CW dataset and UAV images on the ALL-DATA-UAV-CW dataset. The following set of hyperparameters provide the most appropriate result concerning size, runtime, and performance of RF-OBJ and RF-KP in the case of UGV and UAV data: number of trees of 100, number of random features of 25, depth of trees of 10, keypoint size of [10 × 10], and lattice distance of 10. Note that the keypoint size and the lattice distance are only used in the keypoint-based approach. This configuration of hyperparameters leads to an average F1-score of around 84 % across the classes crop and weed. Note that we fix this set of hyperparameters for UGV and all UAV experiments in the remainder of this section.

6.3.1.2 Aggregation of RF-KP and RF-OBJ into RF-CAS

Our proposed RF-CAS classification system represents the base classification model for the purely visual plant classification based on handcrafted features using random forests. The RF-CAS approach is a cascade of the two subordinated approaches RF-KP and RF-OBJ.

Our keypoint-based approach RF-KP has the advantage that it can deal with plants that overlap in image-space but at the cost of being computationally expensive. In contrast, our object-based approach RF-OBJ has the advantage that it is substantially faster than the keypoint-based approach but cannot deal with overlapping crop plants and weeds. Thus, we combine both approaches in a cascade to explicitly exploit their respective advantages. To show the advantage of aggregating RF-KP and RF-OBJ in a cascade, we separately evaluate the respective classification performance of RF-KP, RF-OBJ, and finally, of RF-CAS.

In sum: we first show that both the keypoint-based classifier RF-KP as well as the object-based classifier RF-OBJ perform well on real-world datasets and can be combined to compensate their respective drawbacks. Second, we that the RF-CAS approach is suitable for classifying crop plants and weeds under challenging real-world conditions, including a substantial amount of overlapping plants and weeds with an average processing rate of around 9 Hz. The comparison of RF-KP and RF-OBJ is designed to support our first claim above. We analyze the quality of their respective classification outputs to motivate the need for a combined approach RF-CAS.

In the evaluation reported below, we use a subset of images from the STUTT-CW-15 dataset. We call this subset STUTT-CW-15-SUB, which poses challenging conditions

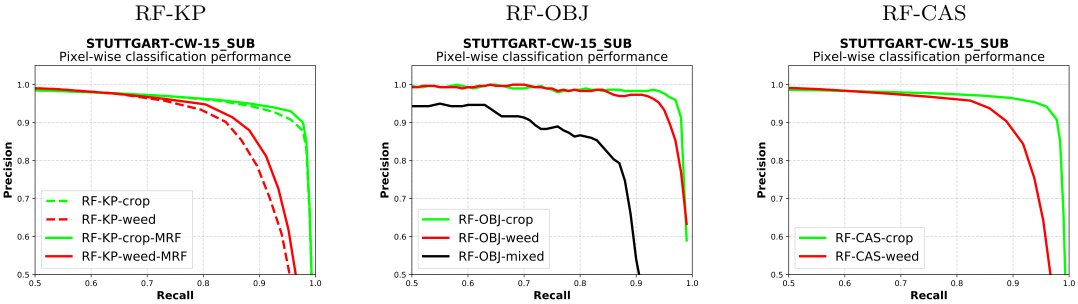


Figure 6.4: Precision-recall curves for the classification performance on the STUTT-CW-15-SUB dataset. Left: Performance of the keypoint-based approach RF-KP. Middle: Performance of the object-based approach RF-OBJ. Right: Performance of the cascaded approach RF-CAS. For a more detailed view on the results, we show only a cropped area of precision-recall space, i.e. the best 50 %.

concerning the downsides of RF-KP and RF-OBJ. It consists of 1,718 images and represents challenging conditions in terms of weed pressure and inter-class overlap, i.e., overlapping sugar beet plants and weeds. Figure 3.10 illustrates some representative example images reflecting these conditions. We show the RGB+NIR images along with the corresponding pixel-wise ground truth information.

Regarding the class labels for analyzing the classification results we refer to sugar beet plants (ω_c, green) as crop and weeds (ω_w, red) as weed in this section. For the evaluation of the object-based classification RF-OBJ, we additionally refer to mixed objects as ω_m (black). For all experiments in this section, we define an object as a mixed object if it consists of both sugar beet and weed pixels, and both classes contribute with more than 10 % of the total pixels each.

We train the random forest of the RF-KP and RF-OBJ according to the set of hyperparameters that we evaluated in the previous section. We train the approaches on the same random 75 % training split of the STUTT-CW-15-SUB dataset, deploy it on a 20 % test split and use the remaining 5 % as validation data. Note that in this section, we do not use any features about the spatial arrangement of the plants.

The resulting precision-recall curves are depicted in Figure 6.4. They illustrate the achieved pixel-wise classification performance on the STUTT-CW-15-SUB dataset for the crop and weed. Note that we intentionally do not present the performance for the soil class, as it solely reflects the performance of the vegetation classification in the case of the random forest-based approaches. In the case of the keypoint-based approach, the term MRF refers to the random forest combined with the MRF, as described in Section 4.3.5.

For the keypoint-based classification, we achieve maximum overall accuracy of 96 % at $t = 0.5$, i.e., labeling with the most likely class according to Equation (2.23). This means that we do not adjust the keypoint-based approach to have a preference for sugar beet or weed. For $t = 0.5$, we achieve a recall of 95 % for sugar with a precision of 95 %. In terms of weeds, we obtain a recall of 85 % with a precision of 84 %. As the recall for sugar beet is 95 %, the system classifies the majority of plants correctly and keeps the percentage of sugar beet keypoints, which are wrongly classified as a weed, small with 5 %. The corresponding precision-recall plot manifests that nearly all vegetation pixels, which are classified as sugar beet, are predicted with high confidence. In terms of runtime, RF-KP analyzes the images on average with a processing rate of around 1 Hz.

In terms of MRF smoothing, we gain an improvement of 3 % in overall accuracy. The main reason for that increase in precision for weeds is due to the smoothing of stem regions, as depicted in Figure 4.9. These results show that the classification results take advantage of the spatial smoothing of the class labels.

The purely object-based approach RF-OBJ can select among the three classes $\{\omega_c, \omega_w, \omega_m\}$ using $\omega = \operatorname{argmax}_\omega p(\omega \mid \Phi(\boldsymbol{V}, \Theta))$. We classify 97 % of the area covered by sugar beet plants correctly with a precision of 95 %. For weeds, we obtain a recall of 95 % with a precision of 95 %. Even most of the mixed object are classified correctly with a recall of 86 %, but with a lower precision of 84 % compared to crops and weeds. This lower performance is due to the substantially smaller number of training examples for the mixed class. In STUTT-CW-15-SUB, the mixed objects cover around 11 % of the vegetation. Because of this, the overall performance of the object-based approach is limited. In order to deal with overlapping plants, both classification systems should be combined.

For the cascaded RF-CAS approach, we report the results with a minimum probability of $t^{\mathcal{O}}_{min} = 0.5$ in Equation (4.18). Thus, we pass objects that are predicted with a lower probability $t^{\mathcal{O}}_{min} < 0.5$ to the keypoint-based approach. Finally, RF-CAS achieves a recall of 97 % for sugar beets with a precision of 95 %. In terms of weeds, we obtain a recall of 90 % with a precision of 98 %. Generally, the RF-CAS classifier offers a similar performance as for the keypoints. The main differences are a better precision for weeds, which arises from high precision for weed of the object-based classification and substantially faster execution time of around 10 Hz. The RF-CAS approach is faster, as it processes most of the vegetation in the image with the object-based approach. Thus, only a few parts of the image are passed through the slower keypoint-based approach.

Figure 6.5 depicts two example classification results on the STUTT-CW-15-SUB dataset obtained under challenging situations with substantial weeds growing close to sugar beets. These results indicate the potential of the cascaded classification. The object-based classification performs almost perfectly, meaning that single sugar beets, weeds, as well as mixed objects, are classified correctly.

The keypoint-based classification (Figure 6.5, left column) can separate the sugar beet plants from the adjacently grown weeds but still fails to predict a few keypoints correctly. However, as a stand-alone classification system, the object-based approach is not able to separate whole vegetation into crops and weeds (if no mixed labels are allowed). RF-CAS achieved the best classification performance. Further experiments supporting our made claims in this section can be found in [87].

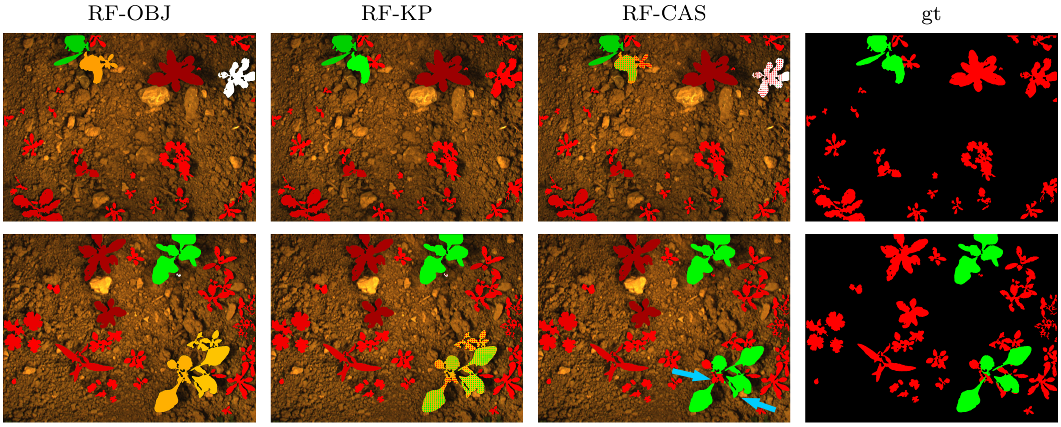


Figure 6.5: Visual illustration of results (one result per row), achieved by the combined classification approach. From left to right: keypoint-based, object-based, and combined classification including mixed (orange) and uncertain (white) objects, combined classification at full resolution, ground truth. The confidences for crop plants (green) and weeds (red) are encoded with the radius of the keypoints or in case of objects with the color intensity. Drawn arrows (blue) indicate classification errors.

6.3.1.3 Feature Importance

Next, we design this analysis to investigate which visual features are important for the random forest-based crop-weed classification task. Therefore, we train the RF-KP and RF-OBJ approaches on the entire ALL-DATA-CW dataset and use the internal out-of-bag estimates of the classifier to measure the feature importance according to Breiman [15]. The key idea is that each particular feature that is present in a constructed tree is permutated, and a classification is performed for the out-of-bag data points under the permutation. A feature is considered to be more important, the more its permutation affects the classification performance.

The ten most important features for both, the keypoint-based as well as the object-based approach, are listed in Table 6.2. As can be seen, the NDVI and the hue information, as well as their respective gradient and texture representations, are key supporters for the keypoint-based classification task. For the object-based classification, the most relevant features are also related to the NDVI information. Furthermore, the shape features appear highly relevant, probably, as they describe mostly complete plants in these settings. For both approaches, the best 50 features (15 % of all used features), hold approximately 44 % of the overall feature importance.

Without showing the results in detail, we also performed the same experiment on the STUTT-CW-15 dataset. The ranking of particular features changes regarding Table 6.2. However, the NDVI, hue, and lightness appear again under the top ten important features.

Table 6.2: The ten most expressive visual handcrafted features for the training on the ALL-DATA-CW dataset. We present the features for keypoint-based classification as well as for the object-based classification. See Table 4.1 for an explanation of the features.

Rank	Keypoint feature	Object feature
1	$\mathcal{V}_9(\nabla\mathbf{I}_H)$	$\mathcal{V}_4(\nabla\mathbf{I}_{NDVI})$
2	$\mathcal{V}_7(LBP(\Delta\mathbf{I}_{NDVI}))$	$\mathcal{V}_8(\nabla\mathbf{I}_{NDVI})$
3	$\mathcal{V}_{17} \rightarrow \mathcal{V}_9\nabla\mathbf{I}_{NDVI}/\mathcal{V}_9\Delta\mathbf{I}_{NDVI}$	$\mathcal{V}_{16}$ Formfactor
4	$\mathcal{V}_4(\Delta\mathbf{I}_H)$	$\mathcal{V}_6(\nabla\mathbf{I}_L)$
5	$\mathcal{V}_8(LBP(\Delta\mathbf{I}_{NDVI}))$	$\mathcal{V}_7(\nabla\mathbf{I}_L)$
6	$\Delta\mathbf{I}_{NDVI}$	$\mathcal{V}_8(\Delta\mathbf{I}_{NDVI})$
7	$\mathcal{V}_5(\nabla\mathbf{I}_L)$	$\mathcal{V}_6(\Delta\mathbf{I}_{NDVI})$
8	$\mathcal{V}_7(\nabla\mathbf{I}_H)$	$\mathcal{V}_3(LBP(\Delta\mathbf{I}_{NDVI}))$
9	$\mathcal{V}_6(\nabla\mathbf{I}_L)$	$\mathcal{V}_{14}$ aspect ratio
10	$\mathcal{V}_9(LBP(\mathbf{I}_{NDVI}))$	$\mathcal{V}_3(LBP(\Delta\mathbf{I}_G))$

6.3.2 Fully Convolutional Neural Network Model

The experiments in this section are designed to evaluate a set of hyperparameters for our proposed fully convolutional network classifier that we call FCN. We tune the hyperparameters and compare our proposed FCN architecture with other state-of-the-art architectures to show that our proposed model provides superior overall performance.

6.3.2.1 Hyperparameter Search for the FCN Architecture

Bengio [10] provides a list of the most crucial hyperparameters to tune in the case of fully convolutional neural networks. In the course of writing this thesis, we have built up years of experience in dealing with various classifiers and influences of hyperparameter selection. Based on this experience and in line with Bengio [10], we search for the following set of optimization specific hyperparameters: (i) initial learning rate, (ii) learning rate schedule, (iii) batch size, and (iv) the used optimizer.

In terms of the architectural design of the FCN, we modulate the following architectural hyperparameters: (i) number of learnable parameters, (ii) order of layers within the convolutional layer, i.e., 2D convolution - ReLU - batch normalization and batch normalization - ReLU - 2D convolution, (iii) the effectiveness of dropout, (iv) using or avoiding the bias parameter subsequent to the 2D convolution, and (v) the use of bottleneck layers.

In the following, we will discuss influences and evaluate values for specific hyperparameters. For all experiments in this section, we use the average F1-score across all classes as the criterion for evaluating the performance of a particular hyperparameter. We use the same data as for the random forest model described in Section 6.3.1.1.

Architectural hyperparameters We design these experiments to evaluate a set of architectural hyperparameters for our FCN approach. One of the essential characteristics of network architecture is its size. The size of the network is correlated with the capacity of the network, i.e., the ability to encode information. Sufficient capacity is necessary to solve complex classification problems, such as the pixel-wise classification of similar objects. Thus, the network size is directly related to the classification performance. A further connection of the network size exists with the expected runtime for the classification. Roughly speaking, the more extensive the network, the more time the network takes to complete the task and for its training, as more operations need to be calculated. A larger size of the network, however, causes a slower inference time during deployment. Additionally, larger networks typically require more training data to be trained and to reduce the risk of overfitting to a small set of training examples.

We split the network size into its width and depth components. The number of consecutive convolutional layers defines the network depth. The more layers with non-linear units are connected in series, the more descriptive features the model can extract to support classification. The depth of our FCN architecture, which we present in Section 5.2.1, is again controlled by the number of consecutive dense blocks that are always followed by a downsampling operation. Thus, the depth-relevant hyperparameters of the network are is the number of convolution layers within a dense block. In addition to that, we use these parameters to control the network's receptive field, which is responsible for the size of the image content used to classify a pixel in the output.

The width of a network is given by the number of parameters of the layers in the network. Thus, the width is defined over the number of feature maps and the used kernel size for the convolutions. Thus, concerning our DenseNet-based FCN approach, we control its width by setting the growth rate for dense blocks, and the number of feature maps in the first convolutional layer. Note that the number of feature maps in all other layers, except the last layer in the network, is determined by the internal network logic described in Section 5.2.1.

We performed several experiments varying the parameters for the depth and width of our propose FCN architecture, i.e.,

$$\begin{aligned}\text{Kernel size} &\in \{[3 \times 3], [5 \times 5], [9 \times 9], [15 \times 15], [25 \times 25]\}\\ \text{Feature maps in first layer} &\in \{8, 16, 32, 64\}\\ \text{Growth rate} &\in \{2, 4, 8, 16\}\\ \text{Number of layers in dense block} &\in \{2, 3, 4, 5, 6\}\\ \text{Number of stacked dense blocks} &\in \{2, 3, 4, 5, 6\}\end{aligned}$$

We analyze the performance of the FCN approach for UGV images on the ALL-DATA-CW dataset and UAV images on the ALL-DATA-UAV-CW dataset. The following architecture provides the most appropriate result concerning model size, runtime and performance in the case of UGV data: kernel size of $[5 \times 5]$, 32 feature maps in the first layer, the growth rate of 4, number of layers in a dense block of 3, and number of stacked dense blocks of 3. Given this set of hyperparameters, our model has around 178,000 trainable parameters. This configuration of hyperparameters leads to an average F1-score of around 89 % across the classes crop, weed, and soil on the ALL-DATA-CW dataset. In the experiment, there were five further configurations of hyperparameters, which delivered a better performance of about 90-91 % average F1-score. However, all these models have more than six times the number of parameters. Therefore, they are more memory intensive and too slow in terms of runtime. Note that we fix this set of hyperparameters for UGV experiments in the remainder of this section.

For the case with UAV images, we use almost the same architectural design as for UGVs, but with the following exceptions: kernel size of $[5 \times 5]$ and number of stacked dense blocks of 4. Given this set of hyperparameters, our model has around 340,000 trainable parameters. The increased kernel size and additional dense block followed by an additional downsampling operation lead to an increase of the network's receptive field. This change leads to a performance gain of around 5 % in terms of average F1-score for the ALL-DATA-UAV-CW dataset. We argue that the larger receptive field enables the network to learn features considering the more spatial context from the input images, thus enabling the network to extract features describing the relative arrangement of the plants in the field. Note that we fix this set of hyperparameters for UAV experiments in the remainder of this section.

We evaluate hyperparameters that are specific to our sequential FCN-SEQ approach in Section 6.6.3.

Optimization-specific hyperparameters We design these experiments to evaluate a set of hyperparameters for our FCN approach that is specific to its training procedure. As the first two hyperparameters, we test the initial learning rate and the learning rate schedule. Changing the initial learning rate is about testing, in which the learning rate maximizes the learning progress of the fully convolutional neural network. Therefore we tested under which learning rate $LR \in \{1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ the fully convolutional neural network provides the best-performance when training for a particular number of epochs. We set the number of epochs to $E = 5$. We achieve the best result with a distance of about 4 % average F1-score to the second-best performance for a learning rate of $LR = 0.01$. Next, we evaluate the learning rate schedule. Therefore, we start with LR and decrease the value as training progresses. Therefore, we tested different types, including a step-wise decay, cyclic learning rate, and a manual schedule. We trained the FCN for $E = 10$, respectively, and found no notable difference in performance. However, decreasing the learning rate by a schedule provides a gain of 5 % in average F1-score compared to the training with a constant LR. Thus, we set the initial learning rate to $LR = 0.01$ and divide it by ten after 50,000 performed training steps, i.e., batches. We fix the aforementioned initial learning rate schedule for all experiments we perform in this chapter. We also fix these hyperparameters for the experiments in this section.

Next, we evaluate the effect of the used batch size on the performance of the FCN. The key idea is that a larger batch size leads to more stable training, as the gradients of the loss function can be estimated more accurately. However, a larger batch size comes at the cost of a slower runtime for the training. The batch size further is limited due to the memory capacity of the used hardware. We tested several batch sizes ranging from 1-16 on the ALL-DATA-CW and ALL-DATA-UAV-CW datasets and found that we achieve the best performance with a batch size of 4. This value represents the best trade-off between training time and performance, given our model and datasets.

Finally, we evaluate the effect on the performance when using different optimizers for the training such as stochastic gradient descent, RmsProp [144], and Adam [63]. Therefore, we train the FCN approach for ten epochs using the respective optimizers. We found that Adam and RmsProp lead to a better performance than for the training with SDG. As the performance of RmsProp and Adam are somewhat similar, we pick Adam for all further experiments throughout the experimental evaluation in this chapter.

6.3.2.2 Comparison of FCN with State-of-the-Art Architectures

We design this experiment to show that our proposed FCN architecture provides better performance compared to other state-of-the-art architectures to show that our proposed model provides superior overall performance.

There exist many network architectures providing state-of-the-art performance among different tasks and applications [52, 55, 58, 124, 125, 132].

Our initial approach to plant classification with convolutional neural networks was to try out some of these approaches for our application. However, we found that the native implementations of these approaches did not deliver the desired performance immediately. This fact has led us to develop our network architecture, adapted to our specific needs. The product of these efforts manifests itself in our proposed fully convolutional neural network architecture FCN.

To verify that our FCN architecture is the right choice in regards to the plant classification task, we design the following experiment for a performance evaluation of FCN against commonly deployed state-of-the-art architectural designs, i.e., Resnet-34 [52], DarkNet [124], ErfNet [125], MobileNet-V2 [132], and DeepLab-V3+ [22].

Regarding the task of pixel-wise classification, also called semantic segmentation, we conceptually separate a network architecture into a feature extractor, often called the backbone or encoder network, and a decoder network. As encoders, we use the network architectures of Resnet-34, DarkNet, ErfNet, and MobileNet-V2. All these approaches have been investigated in several studies, and it has been shown that these architectures are particularly suitable for the extraction of useful features for various tasks such as image classification, object detection, and semantic segmentation, i.e., pixel-wise classification of entire images. A decoder of network architecture, however, is task-specific. Therefore, we use the best performing decoder DeepLab-V3+, which is known to us at the time of conducting these experiments. We implemented the FCN approach using the Tensorflow library for deep learning [2]. All other architectures are implemented within an internal modification of the Bonnet package [98], which is an open-source training and deployment framework for pixel-wise classification in robotics that builds upon the PyTorch library [109].

We evaluate all architectures regarding their performance in challenging conditions and under changing field conditions. First, we use the ALL-DATA-CW dataset for evaluation, as it poses challenging conditions for the classifiers due to its broad internal diversity regarding different growth stages of plants, weed types, and soil conditions. We run the training on a 70 % training split, test on a 25 % split, and use the remaining 5 % for evaluation according to Section 6.2.2.

Second, we train the models on 95 % of the BONN-CW-16 dataset and test on STUTT-CW-15, BONN-CW-17, ZURICH-CW-16, and ANCONA-CW-18 datasets. We average the performance for the test datasets and refer to it with ALLOTHER-CW. The achieved performance on ALLOTHER-CW reflects the generalization capabilities to a new field environment. Note that for the FCN approach, this experimental setup is the same as in our crop-weed classifications experiment in Section 6.5. For all architectures, we use the same preprocessing procedure as described in Section 4.2 and initialize the weights a truncated normal distribution as proposed by [51].

We train the models using the Adam optimizer with a mini-batch size of $B = 4$ and use a weighted cross-entropy loss according to Equation (2.15), where we penalize prediction errors for the crop plants and weeds by a factor of 10. As the optimal learning rate schedule depends on the training data in combination with the network architecture, we first perform an individual hyperparameter search for all baseline architectures and report only the best individually achieved performance. Therefore, we use the same search strategy as proposed for our FCN approach described in the previous Section 6.3.2.1.

Table 6.3 summarizes the obtained pixel-wise classification performance. For the second run, we report the performance when averaging the performance measures overall used test datasets. The results demonstrate the superior classification performance of our proposed FCN architecture under similar and changing field conditions.

Regarding similar field conditions, our FCN approach achieves a gain of around 3 % in terms of average F1-score compared to the second-best architecture, i.e., DarkNet. We make the general observation that all tested baseline architectures have problems with classification in the marginal areas of plants. The plants and weeds are predicted too large in many cases so that a remarkable number of actual soil pixels are also recognized as plant pixels. This leads to a decrease in the precision of the crop and weed classes, as can bee seen in Table 6.3. Our FCN approach provides sharper results in these regions. We associate this pattern to the use of DenseNet-like architecture and the use of skip connection within our approach, see Section 5.2.1. Both these components can lead to better use of features from earlier layers, which often represent differently oriented edges in the image data. This, in turn, can lead to better recognition of the plant edges, which are often characterized by large gradients in the input image.

Also, in terms of changing field conditions, our FCN approach outperforms all

Table 6.3: Comparison of FCN, Resnet-34, ErfNet, and MobileNet-V2. We report the pixel-wise average F1-score for the classes crop, weed, and soil.

Approach	**Average**		
	F1	P	R
Training: ALL-DATA-CW (70 %) Deployment: ALL-DATA-CW (25 %)			
FCN (ours)	**89.0**	**85.0**	**93.8**
Resnet-34	84.0	81.2	84.9
ErfNet	79.7	75.5	82.6
DarkNet	85.1	79.2	90.3
MobileNet-V2	76.3	71.2	80.6
Training: BONN-CW-16 (95 %) Deployment: ALLOTHER-CW			
FCN (ours)	**72.5**	**74.9**	**71.2**
Resnet-34	68.6	75.8	64.8
ErfNet	53.4	64.4	45.3
DarkNet	67.6	71.9	67.8
MobileNet-V2	36.3	42.6	34.4

baseline models achieving a gain of around 4 % in terms of average F1-score compared to the second-best architecture, i.e., DarkNet. This indicates that our approach has slightly better generalization capabilities. Thus, it can exploit the extracted features better when being deployed in new and previously unseen field environments.

6.4 Vegetation Classification

We design the experiments in this section to analyze the quality of our plant classification systems for UGVs regarding their performance for the vegetation classification. The vegetation classification is a pixel-wise classification considering two classes vegetation and soil (background). This binary classification is a key processing step for the random forest (RF-*) based approaches, as the subsequent plant classification step relies on it.

We analyze the performance of the vegetation classification module of the random forest-based approaches, which we present in Chapter 4, and of our proposed FCN approach, which we describe in Section 5.2 of Chapter 5. In Section 6.4.1, we evaluate the threshold-based vegetation classification performance exploiting different vegetation indexes to quantify, which of these indexes is the best image representation for this task. In Section 6.4.2, we analyze the performance of the FCN approach and compare it to the threshold-based vegetation classification. We explicitly evaluate the performance under similar and changing field conditions, as described in Section 6.2.1. Moreover, we evaluate the effectiveness of exploiting additional NIR information by analyzing the performance, when using RGB+NIR images as input or when we solely use RGB images and input.

6.4.1 Threshold-Based Performance Using Different Vegetation Indexes

For the targeted application of robotic weed control, it is crucial for the random forest-based classifiers to achieve a high recall for the vegetation class, as the subsequent feature extraction as well as the plant classification only further analyze the predicted vegetation pixels. To analyze which vegetation index serves as suitable representation for the vegetation classification, we evaluate commonly applied vegetation indexes as the basis for the threshold-based vegetation classification. To this end, we analyze the normalized difference vegetation index given in Equation (6.1), the excess green index given in Equation (6.2), the triangular greenness index given in Equation (6.4), the normalized difference index given in Equation (6.3), the excess green minus excess red index given in Equation (6.5), and the color index of vegetation extraction given in Equation (6.6):

$$NDVI = \frac{NIR - R}{NIR + R} \tag{6.1}$$

$$ExG = 2G - R - B \tag{6.2}$$

$$NDI = \frac{G - R}{G + R} \tag{6.3}$$

$$TGI = G - 0.39\,R - 0.61\,R \tag{6.4}$$

$$ExGR = ExG - 1.4\,R - G \tag{6.5}$$

$$CIVE = 0.881\,G - 0.441\,R - 0.385\,B - 18.787 \tag{6.6}$$

All these vegetation indexes seek to transform the input image into a bimodal index distribution emphasizing the vegetation pixels through high values and the soil pixels through low values. Figure 6.6 illustrates example images and the evaluated vegetation indexes, respectively. Note that we scale and shift the values of all vegetation indexes to the range of $[0, 255]$ and convert them into an 8-bit representation. Further descriptions of the indexes can be found in Hamuda *et al.* [47] and Torres-Sanchez *et al.* [147].

The goal is to compute a threshold $t \in [0, 255]$ to obtain a binary mask separating the vegetation from the background according to Equation (4.3). First, we manually search for t by iterating over the entire $t \in [0, 255]$ range and select the threshold that provides the best classification performance considering the ground truth for a training dataset. We use a random 10 % split of the BONN-CW-16 dataset as training data for the search of t. Through the manual search, we imitate a human operator that selects an appropriate threshold for the vegetation classification task given the data.

Second, we automatically derive t by using Otsu's method [104]. Otsu's method can be seen as a one-dimensional case of the discriminant analysis of Fischer [32], i.e.,

$$Q(t) = \frac{\sigma^2_{\omega_{v,b}}(t)}{\sigma^2_{\omega_v}(t) + \sigma^2_{\omega_b}(t)} \quad \text{with} \quad t \longrightarrow max(Q). \tag{6.7}$$

Here, ω_v refers to the vegetation class and ω_b to the background class. The algorithm derives t iteratively. It assumes a bimodal Gaussian distribution and that both

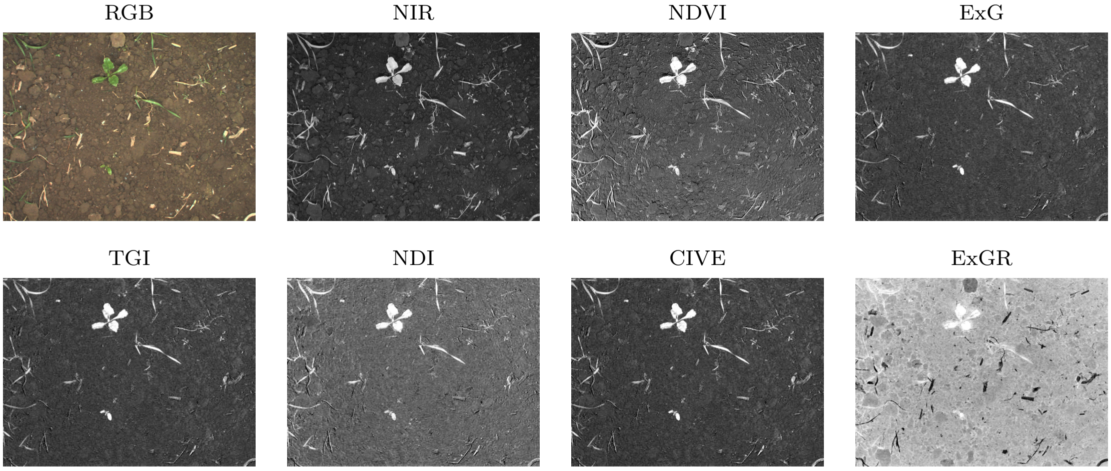


Figure 6.6: Visualization of different vegetation indexes. Note that only the NDVI exploits the additional NIR information.

modes of the distribution refer to the respective mean values of the classes. By employing Otsu's method, we learn a separate threshold for each image in an unsupervised manner. Thus, we do not need access to labeled data.

Best vegetation index Table 6.4 summarizes the obtained performance for the threshold-based vegetation classification. In the case of the manually searched threshold, the NDVI-based method outperforms all other vegetation indexes in terms of F1-score. It provides reliable results with an average F1-score of >90 % on every dataset. Averaging the performance across all used datasets, we achieve around 94 % average F1-score. Thus, it classifies the majority of the vegetation and background pixels correctly. With an average F1-score of around 88 % across all datasets, the ExG serves as the most appropriate representation regarding the indexes using the RGB information solely, not requiring NIR information. The obtained average F1-scores for the other analyzed vegetation indexes achieve a lower performance. Based on these results, we rely on the NDVI when having access to NIR information and on the ExG when solely using RGB for the vegetation classification of all random forest approaches.

Figure 6.7 depicts typical results of the obtained vegetation mask $\mathbf{I}_{VMASK}$ obtained through the NDVI- and the ExG-based thresholding. In general, we can observe qualitatively that the NDVI-based approach leads to better results for the classification of vegetation edges, and therefore small objects, such as weeds.

Atomated thresholding Otsu's automated thresholding achieves a comparably poor performance across all datasets. The achieved performance never reaches more than 63 % average F1-score, which is not sufficient for subsequent plant classification. We qualitatively inspect the results and conclude that the poor performance of Otsu's method is mainly caused by the imbalance in the occurrence of the two classes. On average vegetation, pixels occur with 2 %, whereby the background constitutes the re-

Table 6.4: Pixel-wise vegetation classification performance obtained through the threshold-based approach using different vegetation indexes. We either obtain the threshold t by manual search or estimate it by Otsu's method [104]. We report the average F1-scores across the classes vegetation and soil.

Vegetation index	NDVI	ExG	TGI	NDI	CIVE	ExGR
Manual search on: BONN-CW-16 (10%)						
90% of BONN-CW-16	95.6	93.6	86.1	72.2	68.8	78.0
BONN-CW-17	92.7	89.7	91.0	78.9	80.6	82.2
STUTT-CW-15	95.1	92.0	75.2	66.6	63.5	72.5
ANCONA-CW-18	95.3	80.5	69.9	57.9	64.6	62.0
ZURICH-CW-16	90.1	83.1	81.6	69.1	73.7	81.5
Average	93.8	87.8	80.8	68.9	70.2	75.2
Otsu's method [104]						
BONN-CW-16	58.1	85.6	68.9	39.9	58.0	49.8
BONN-CW-17	82.5	71.7	68.3	65.3	70.1	62.8
STUTT-CW-15	49.1	45.2	55.1	42.0	42.5	56.5
ANCONA-CW-18	61.7	48.9	66.4	51.6	60.0	52.8
ZURICH-CW-16	45.6	59.4	53.2	45.9	45.9	67.3
Average	59.4	62.2	62.4	48.9	55.3	57.8

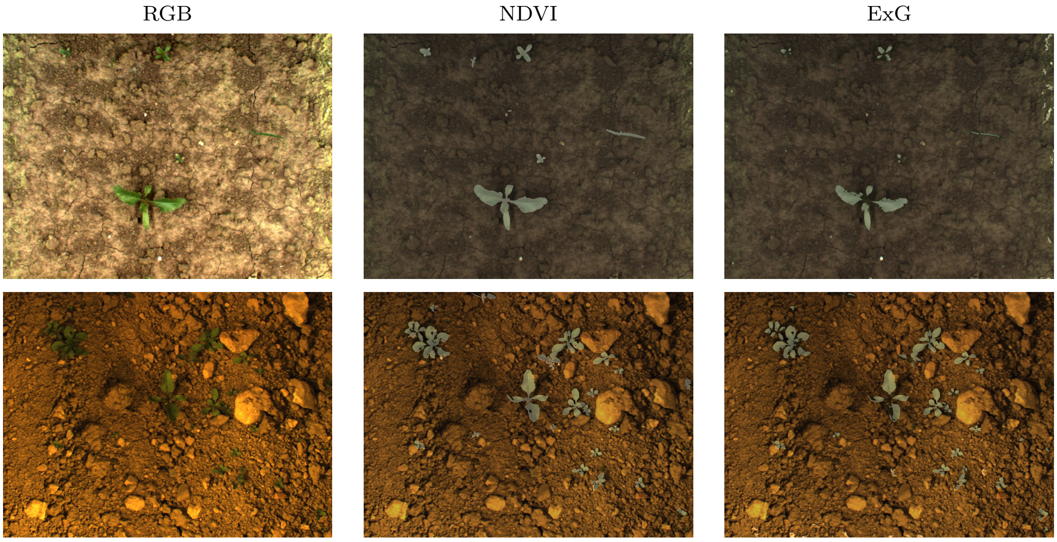


Figure 6.7: Typical results for vegetation mask $\mathbf{I}_{VMASK}$ obtained through the NDVI- and the ExG-based thresholding. We show an overlay of the mask (white) on top of the RGB image.

maining 98 %. Weed control has to be performed early in the crop season, where this ration can be even more imbalanced.

We conclude that Otsu's method is not a reliable option to derive *t* for the vegetation detection on our data. Therefore, we next investigate vegetation classification using our neural network approach and show that by this, we are capable of reliably classifying vegetation even under changing field conditions.

6.4.2 Comparison of Threshold-Based and Fully Convolutional Neural Network-Based Vegetation Classification

In this experiment, we compare the performance of the threshold-based vegetation classification using the NDVI and the ExG index with the one obtained by our proposed FCN and its RGB-only variant FCN-RGB. In contrast to the threshold-based approaches, FCN and FCN-RGB learn a rather sophisticated model to solve the task. We train the fully convolutional neural network models on the same random 10 % split of the BONN-CW-16 dataset, deploy it on an 85 % test split, and use the remaining 5 % as validation data to perform early stopping as described in Section 6.3.2.1. This setup reflects the performance under similar field conditions. To furthermore analyze the generalization capabilities to new and unseen field environments, we also deploy the trained model on the BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16 datasets.

Table 6.5 summarizes the obtained pixel-wise performance for the threshold-based and fully convolutional neural network vegetation classification. First, we compare FCN with FCN-RGB. Under similar as well as under changing field conditions, both approaches perform on the same level with an average F1-score of 91 % across all datasets (changing and similar). This result is remarkable because the RGB-only variant FCN-RGB only needs the RGB input to achieve the same performance as with exploiting the additional NIR information. This pattern is different in the case of the threshold-based approaches. Here, the NDVI-based thresholding outperforms the ExG-based variant with around 6 % average F1-score across all datasets. Thus, concerning the vegetation classification, the fully convolutional neural network approaches better exploit the RGB and compensate for the additional NIR information.

Second, we compare FCN to the NDVI-based thresholding. Here the performance is mostly comparable with each other on the BONN-CW-16 dataset, i.e., when operating under similar field conditions. The FCN achieves a pixel-wise average F1-score of around 97 % and the NDVI-based thresholding 96 %. By looking at the pixel-wise performance, the FCN approach provides a solid recall of 99 % at a precision of 88 %, which means that 99 % of the actual vegetation pixels are classified correctly. Here, the NDVI-based thresholding approach achieves a recall of 94 % at a precision of 90 %. This means that the FCN approach can detect a larger portion of the actual present vegetation in the data than the threshold-based approaches. The higher recall of the

Table 6.5: Pixel-wise vegetation classification performance obtained by using the NDVI and ExG vegetation indexes for the threshold-based classification as well as obtained through our FCN and FCN-RGB approaches. We report the class-wise and average F1-score (F1), precision (P) and recall (R) according to Equation (2.23) in percent. For the vegetation class, we additionally report the object-wise performance.The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Vegetation**						**Soil**
	F1	P	R	F1_{px}	P_{px}	R_{px}	F1_{obj}	P_{obj}	R_{obj}	F1
Training: BONN-CW-16 (10%) Deployment: BONN-CW-16 (85%)										
NDVI	95.6	94.6	96.8	91.4	89.2	93.8	94.8	90.4	99.7	99.8
ExG	93.6	94.8	92.6	87.5	89.8	85.3	86.5	89.9	83.4	99.7
FCN (ours)	96.6	93.9	99.7	93.3	87.8	99.5	97.4	95.5	99.3	99.8
FCNRGB (ours)	96.0	92.9	99.6	92.2	85.9	99.4	96.8	95.3	98.5	99.8
Training: BONN-CW-16 (10%) Deployment: BONN-CW-17										
NDVI	92.7	89.3	96.8	85.8	78.7	94.2	90.2	82.2	99.9	99.5
ExG	89.7	94.9	85.7	79.8	90.4	71.5	87.4	91.0	84.0	99.6
FCN (ours)	91.2	86.0	98.5	83.1	72.1	98.2	92.7	87.0	99.2	99.3
FCNRGB (ours)	91.8	87.6	97.3	84.2	75.3	95.6	90.9	86.1	96.2	99.4
Training: BONN-CW-16 (10%) Deployment: STUTT-CW-15										
NDVI	95.1	94.9	95.4	90.4	90.0	90.9	93.8	91.2	96.6	99.8
ExG	92.0	91.6	92.4	84.2	83.4	85.1	88.8	85.2	92.8	99.7
FCN (ours)	94.2	90.5	98.8	88.7	81.0	97.9	90.5	84.6	97.2	99.7
FCNRGB (ours)	92.7	88.4	98.2	85.6	76.8	96.8	87.9	81.8	94.9	99.7
Training: BONN-CW-16 (10%) Deployment: ANCONA-CW-18										
NDVI	95.3	94.1	96.5	90.6	88.3	93.1	93.4	90.2	96.9	99.9
ExG	80.6	75.6	88.1	61.4	51.2	76.6	63.9	55.0	76.1	99.7
FCN (ours)	93.6	89.1	99.2	87.2	78.3	98.4	91.4	86.4	97.0	99.9
FCNRGB (ours)	94.1	91.4	97.3	88.4	82.9	94.6	89.1	85.8	92.6	99.9
Training: BONN-CW-16 (10%) Deployment: ZURICH-CW-16										
NDVI	90.1	86.1	95.3	80.5	72.2	91.0	86.2	76.0	99.6	99.7
ExG	83.1	92.5	77.2	66.5	85.2	54.5	69.8	86.1	58.7	99.8
FCN (ours)	80.5	72.3	98.6	61.4	44.7	97.8	80.8	68.3	99.0	99.6
FCNRGB (ours)	81.1	73.9	94.9	62.5	47.8	90.2	77.3	69.3	87.4	99.7

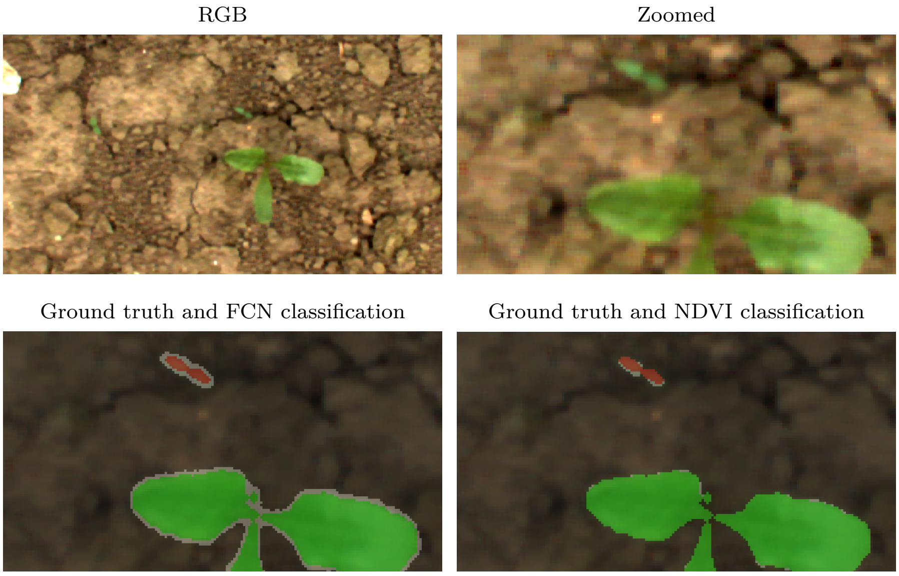

RGB
Zoomed
Ground truth and FCN classification
Ground truth and NDVI classification

Table 6.6: Illustration of the pixel-wise classification error (gray). Compared to the NDVI-based thresholding, our FCN approach tends to produce more false positives in boarder regions of the vegetation, i.e., it predicts pixels as vegetation that actually belong to the soil class. Thus, the vegetation classification of fully convolutional neural network are more "blobby" compared to the NDVI-based thresholding.

FCN, however, comes at the cost of predicting more false positives, i.e., pixels that are predicted as vegetation but belong to the soil class. The achieved recall suggests that this pattern holds for both the vegetation classification under similar field conditions on the BONN-CW-16 dataset and under changing field conditions in new and previously unseen field environments on the BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16 datasets.

Comparing the performance under changing field conditions, we observe a tendency towards a better classification performance for the NDVI-based thresholding in terms of the average F1-score. On average, the threshold-based approach achieves a 7 % better F1-score for the vegetation. This gain is mainly caused by the difference in their performance for vegetation on the ZURICH-CW-16 dataset. Here, the FCN approach obtains around 20 % less in terms of average F1-score compared to the NVDI based thresholding. This result is due to the reduced precision for the vegetation class of around 44 %. The ZURICH-CW-16 dataset is very challenging, as it contains mostly tiny plants of a size ranging from $0.15 - 0.5\,\text{cm}^2$. This also affects the performance of the RGB-only approaches, as we see in terms of the achieved recalls.

6.4.3 Difference of Pixel-Wise and Object-Wise Classification Performance

Figure 6.6 depicts the reason for the systematically lower pixel-wise precision of the fully convolutional neural network. They tend to predict vegetation objects through a more “blobby” shape compared to the NDVI-based thresholding approaches. This systematic leads to more false positives, i.e., vegetation pixels, which are soil pixels, at the edges of the plants. Consequently, the fully convolutional neural networks achieve a lower pixel-wise precision for the vegetation class.

Analyzing the object-wise performance for the vegetation class, the recall R_{obj} for all approaches on all datasets remains at the same high level than in terms of the pixel-wise recall R_{px}. The interesting point is that the fully convolutional neural network object-wise precision P_{obj} is substantially better compared to the pixel-wise precision P_{px}. On average P_{obj} for the FCN is around 11 % and for FCN-RGB around 9 % higher than P_{px}. We see no effect when comparing this gain for similar and changing field conditions.

This effect is mainly caused by the “blobby” predictions of the fully convolutional neural network approaches, as shown in Figure 6.6. In the case of the object-wise metrics, these blobby predictions do not affect the precision, as an object is classified correctly if a certain overlap is between the ground truth and predicted object is given. Thus, from an application point of view, the fully convolutional neural network approaches perform on the same high level as the NDVI-based thresholding.

6.4.4 Conclusions for the Vegetation Classification Experiments

Based on the so far presented results, we draw the following conclusions:

First, the best performing vegetation index is the NDVI, if RGB+NIR is available, and the ExG, if only RGB data is available. However, to achieve a suitable performance, the threshold has to be selected manually. Otsu’s thresholding fails in a situation with highly imbalanced class occurrences and, thus, is not suitable for the vegetation classification task.

Second, the threshold-based vegetation classification using the NDVI provides the best vegetation classification results if NIR information is available. It performs better than the fully convolutional neural network approaches under changing field conditions, thus, generalizes better. However, for the applications of weed removal, the object-wise metric suggests that the fully convolutional neural networks and NDVI-based thresholding are on the same level.

Third, the threshold-based vegetation classification substantially benefits from the additional NIR information. The fully convolutional neural network vegetation classification performs similar when exploiting NIR or solely using RGB data. Thus, in case we solely have access to RGB data, a fully convolutional approach (FCN-RGB) is the method of choice. It can exploit the RGB signal better compared to the threshold-based approach.

6.5 Vision-Based Crop-Weed Classification

We design the experiments in this section to analyze the quality of the vision-based plant classification systems for the UGV systems regarding their crop-weed classification performance. The crop-weed classification task is a pixel-wise classification considering the classes crop, weed, and soil (background). Regarding the random forest-based classification systems, we evaluate the RF-CAS approach, and regarding the deep learning methods, we evaluate the FCN and FCN-RGB approaches. Note that we do not take the RGB-only variant of RF-CAS into account, as the evaluation for the ExG-based vegetation classification in Section 6.4 suggests that no suitable performance can be obtained for this application.

We perform the experiments in this section in two different variants. The first setup is designed to evaluate the classification performance under similar field conditions (Section 6.5.1). This means that the deployed classifier has seen examples from the same field environment during training. The second setup is designed to evaluate the generalization capabilities of the crop-weed classifiers to changing field environments (Section 6.5.2). This means that the classifier is trained on data coming from a particular field and is then deployed in other field environments.

In this section we use the following crop-weed UGV datasets which we describe in Section 3.2.1: BONN-CW-16, BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16. All these datasets are fully labeled in a pixel-wise manner considering the classes crop, weed, and soil. The BONN-CW-16 dataset represents our primary source of training data. It involves around 12,500 labeled images. We acquired every other dataset on a different field at a different point in time. For the performance evaluation under similar field conditions, we consider the BONN-CW-16 dataset and the ALL-DATA-CW dataset, which is an aggregation of all crop-weed datasets. Here, we train the models on a training portion and test them on a held-out test portion, respectively. For the performance evaluation under changing field conditions, we solely train the models on the BONN-CW-16 data and test them on the other dataset, respectively. For all experiments in this section, we use a 5 % split of the training datasets as validation data for the fully convolutional neural network approaches to perform early stopping.

6.5.1 Performance Under Similar Field Conditions

This experiment is designed to evaluate the performance under similar field conditions explicitly. We train the classification models on the respective 70 % training portions of the training datasets and test them on the 25 % portions, respectively. We use the remaining 5 % portion as validation data to perform early stopping.

Table 6.7 summarizes the obtained pixel-wise crop-weed classification performance for the tested approaches on the BONN-CW-16 and ALL-DATA-CW datasets. These results are obtained when we select the class with the highest probability from the predicted distribution across the class labels. For the sake of brevity, we only report

Table 6.7: Pixel-wise crop-weed classification performance under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for labeling with the most likely class according to Equation (2.23) in percent. The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Crop**			**Weed**			**Soil**
	F1	P	R	F1	P	R	F1	P	R	F1
Training: ALL-DATA-CW (70%) Deployment: ALL-DATA-CW (25%)										
FCN	89.0	85.0	93.8	92.4	88.4	96.7	74.7	66.7	85.0	99.8
FCN-RGB	86.2	81.1	92.9	90.2	84.6	96.6	68.8	59.1	82.3	99.7
RF-CAS	82.0	79.1	85.7	81.1	80.2	82.1	65.3	57.5	75.6	99.5
Training: BONN-CW-16 (70%) Deployment: BONN-CW-16 (25%)										
FCN	89.7	84.6	96.7	94.8	91.5	98.4	74.4	62.5	91.8	99.8
FCN-RGB	89.1	84.2	95.7	94.3	90.8	98.0	73.2	62.0	89.3	99.8
RF-CAS	84.4	79.2	91.9	90.6	86.6	94.9	63.1	51.5	81.3	99.5

the F1-score for the soil class as the values for the precision and recall mostly range from 99.5 %-99.9 %. Thus, it is a nearly perfect classification.

First, we analyze the pixel-wise performance of the fully convolutional neural networks on the ALL-DATA-CW dataset. As this dataset is an aggregation of all crop-weed datasets used in this thesis, it contains the largest variety of field conditions, including different growth stages of the crop plants, several weed types present at different growth stages, and diverse soil conditions. In addition to that, the contributing datasets are acquired under different illumination conditions. With an average F1-score of 86 % obtained by the FCN and 84 % obtained by FCN-RGB, the fully convolutional neural network approaches perform well and on a comparable level. Both approaches achieve a recall of around 97 % for the crop plants and 90 % for the weeds. Thus, they classify most of the actual vegetation correctly. Regarding the weed class, both approaches achieve precision ranging from 48 %-52 %. Figure 6.8 (left) depicts the corresponding precision-recall curves.

We observe an advantage in terms of precision for the FCN approach exploiting the NIR information. We argue that the NIR information enables a better classification of class boundaries. The described effect on the precision is specifically noticeable for the weed class since, on average, the weeds are smaller in the ALL-DATA-CW dataset and therefore have a lower pixel-wise probability of occurrence, in terms of the object-based performance illustrated as a precision-recall curve in Figure 6.8 (right). However, the fully convolutional neural networks achieve high F1-scores of around 96 % for weeds. This difference between the pixel-wise and object-wise metric is caused by the effect of “blobby“ predictions for small weeds, as described in Section 6.4.3. First, the object-wise metric is less affected by falsely predicted pixels on borders of plants and weeds. Second, the dataset contains a large number of small weeds that are correctly predicted concerning the object-wise metric, but cause a larger error in the pixel-wise metric in the case of incorrectly classified object boundaries, see Section 6.4.3.

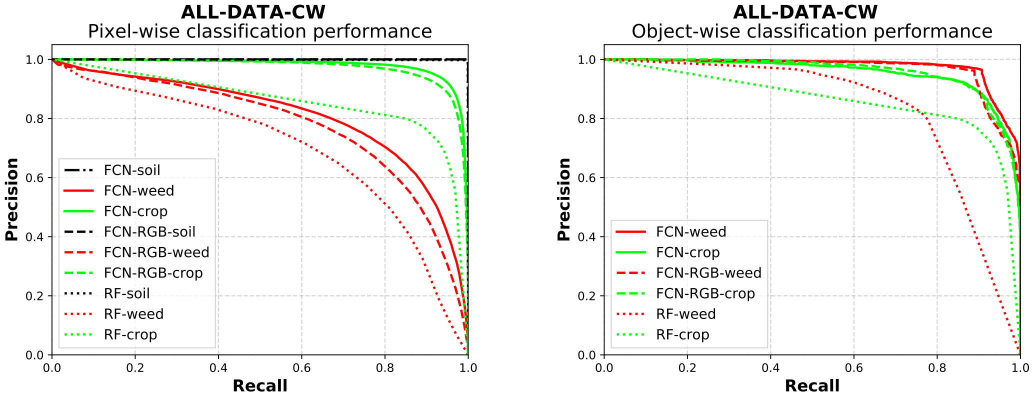


Figure 6.8: Precision-recall curves for the classification performance of the fully convolutional neural network approaches on the ALL-DATA-CW dataset. FCN has a slight advantage for the classification of weeds. For the object-wise performance, the overall performance for both fully convolutional neural network approaches is comparable.

Second, we analyze the performance of the fully convolutional neural networks on the BONN-CW-16 dataset containing sugar beets and weeds observed over an entire season. Here, we find higher comparability of the results in the comparison to the performance on ALL-DATA-CW. Both approaches achieve an average F1-score of around 90 %. This performance is, on the one hand, better and, on the other hand, more similar between FCN and FCN-RGB compared to the performance on ALL-DATA-CW. As the main reason for this, we see the overall lower diversity in the data. As a result, the FCN approach no longer has a notable advantage over FCN-RGB. These results indicate that FCN-RGB can learn discriminative features, which are sufficient for the crop-weed classification task, by solely exploiting the RGB information under similar field conditions.

Third, we analyze the performance of the RF-CAS approach on both datasets. Compared to the fully convolutional neural network, RF-CAS obtains comparably low average F1-scores of around 76 % for the BONN-CW-16 and 69 % for the ALL-DATA-CW dataset. We investigated the confusion matrices according to these results and found that the largest source of error is due to misclassification between classes crop and weed. This statement is also backed by a previous vegetation classification experiment in Section 6.4.2. Here, the NDVI-based vegetation classification that is used within RF-CAS achieves around 94 % average F1-score across all datasets. Thus, most of the confusion goes into the separation of crop plants and weeds. We argue that the learned features do not provide the necessary capacity to cope with the diversity of the visual appearance, the crops plants, weeds, and soil conditions. For clarification, we once again call the performance of RF-CAS on a data set with less intrinsic diversity, namely the experiment on the STUTT-CW-15-SUB data set discussed in Section 6.3.1.2. Here, RF-CAS achieves an average F1-score of around 94 %. Together with the results from this section, this pattern suggests that RF-CAS delivers a worse performance with increasing diversity in the data.

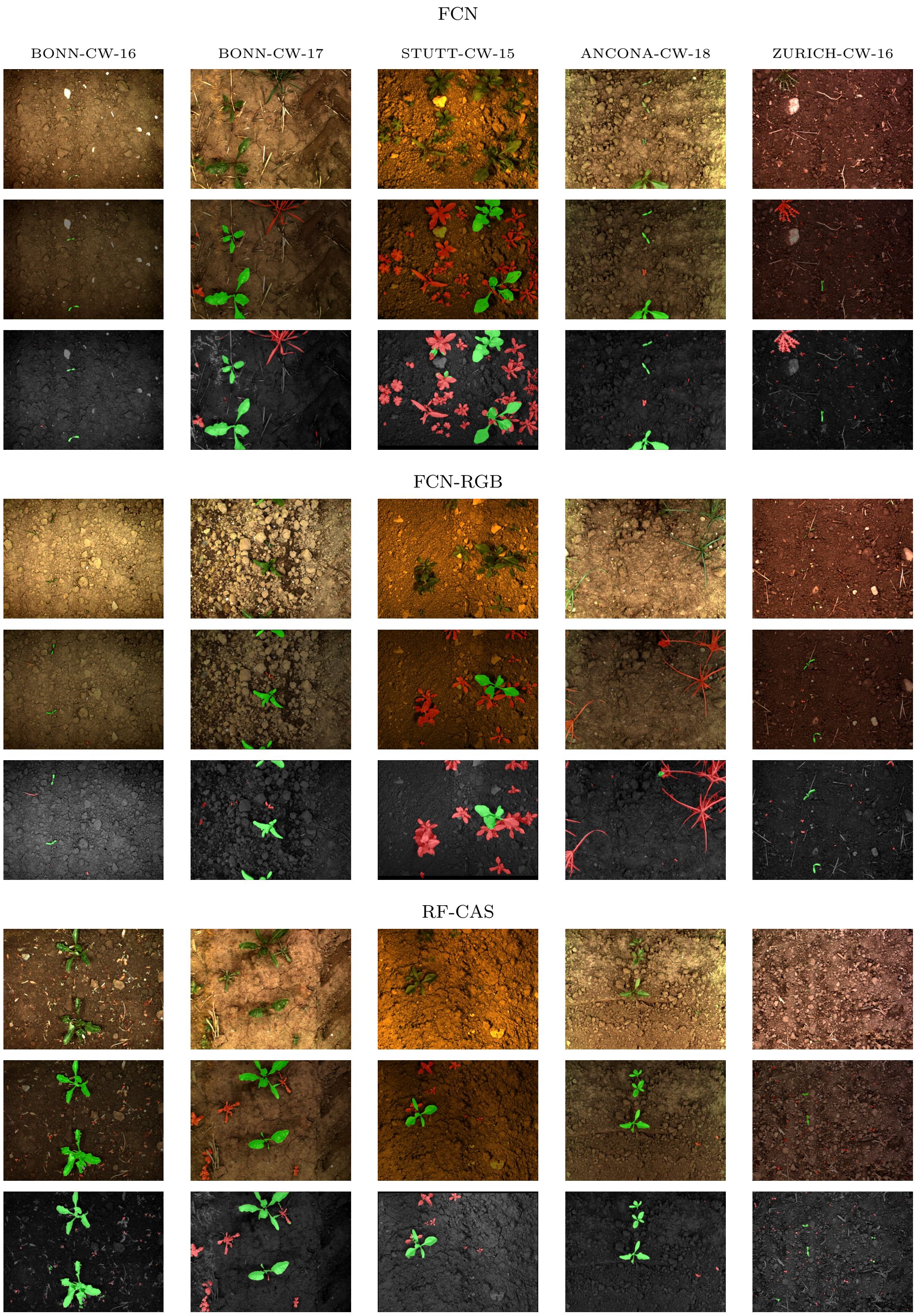


Figure 6.9: Qualitative results for the crop-weed classification performance of FCN, FCN-RGB, and RF-CAS under similar field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on the NIR image. Crop plants (green) and weeds (red) represent the pixel-wise classification.

Finally, we qualitatively analyze the performance of FCN, FCN-RGB, and RF-CAS. Figure 6.9 depicts analyzed images from the experiment on the ALL-DATA-CW dataset. The visual comparison of the fully convolutional neural network and random forest approaches coincides with the quantitative results. The fully convolutional neural network-approaches are better able to identify crop plants and weeds in this experiment. The results on the BONN-CW-16 and ZURICH-CW-16 dataset show that the fully convolutional neural networks properly segment tiny plants, and the results on the STUTT-CW-15 datasets demonstrate their ability to produce correct classifications in situations with high crop-weed overlap. We found that most of the error is due to the wrong classification of vegetation pixels close to class borders. The analysis of RF-CAS shows that this approach mainly has problems in distinguishing between crop plants and weeds and not in terms of separating the vegetation from the soil. Thus, the NDVI-based segmentation works appropriately, whereas the subsequent classification is not capable of providing a suitable crop-weed classification under these conditions.

6.5.2 Performance Under Changing Field Conditions

This experiment is designed to evaluate the performance under changing field conditions explicitly. We train the RF-CAS, FCN, and FCN-RGB approach using a 95 % training portion of the BONN-CW-16 dataset and test on the entire BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16 datasets. We use the remaining 5 % portion of the BONN-CW-16 dataset as validation data to perform early stopping. Through this, we can evaluate the generalization capabilities of the crop-weed classifiers for different field environments. For the application of an agricultural robot in a weed-control scenario, we want to train a “good enough” classifier for the application once and then deploy it on different robots operating in different field environments, where the visual appearance of the crop plants, weeds, and soil cannotably change. Thus, from an application point of view, this experiment reflects practical circumstances for the classifiers.

Table 6.8 summarizes the obtained pixel-wise crop-weed classification performance. For the sake of brevity, we only report the F1-score for the soil class, as the values for the precision and recall mostly range from 99.5 %-99.9 %.

Concerning the performance under similar field conditions, see Section 6.5.1, we observe a substantial loss in performance for all approaches on every test dataset under changing field conditions. In terms of the pixel-wise performance, we achieve an average F1-score of around 68 % by RF-CAS, 71 % by FCN-RGB, and 73 % by FCN averaging all datasets. All tested approaches obtain their best performance on the BONN-CW-17 dataset. All approaches obtain the lowest performance on the ZURICH-CW-16 dataset. Considering the cross-dataset domain shift described in Section 3.2.1, these results are expected because the domain shift between BONN-CW-16 and BONN-CW-17 is the smallest and between BONN-CW-16 and ZURICH-CW-16 is the largest. However, even for the BONN-CW-17 dataset, the obtained recall for the crop class ranges from 52 % to 73 %.

Table 6.8: Pixel-wise crop-weed classification performance under changing field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-CW-16 Deployment: BONN-CW-17										
FCN	78.9	76.4	85.6	78.4	84.6	73.0	58.8	45.1	84.4	99.5
FCN-RGB	76.5	78.2	79.3	70.5	87.5	59.0	59.3	47.4	79.3	99.6
RF-CAS	68.8	68.3	71.4	57.8	65.0	52.0	49.2	40.4	62.8	99.5
Training: BONN-CW-16 Deployment: STUTT-CW-15										
FCN	66.3	63.6	77.3	54.8	60.9	49.8	44.4	30.4	82.6	99.6
FCN-RGB	64.6	67.9	72.6	54.4	77.3	42.0	39.9	27.0	76.2	99.5
RF-CAS	63.3	68.0	60.2	57.3	59.9	55.0	32.6	44.4	25.8	99.8
Training: BONN-CW-16 Deployment: ANCONA-CW-18										
FCN	71.1	68.0	84.2	81.3	83.6	79.2	32.2	20.6	73.5	99.8
FCN-RGB	73.7	70.7	78.5	75.3	75.6	75.1	45.8	36.8	60.7	99.8
RF-CAS	68.3	67.1	70.1	57.2	59.4	55.1	48.1	42.3	55.7	99.6
Training: BONN-CW-16 Deployment: ZURICH-CW-16										
FCN	62.9	54.1	84.1	41.2	29.4	68.9	47.8	33.4	83.8	99.6
FCN-RGB	64.6	58.2	76.6	49.4	42.1	59.7	44.9	32.9	70.5	99.7
RF-CAS	61.5	67.5	60.5	29.3	52.1	20.4	55.3	50.4	61.2	99.9

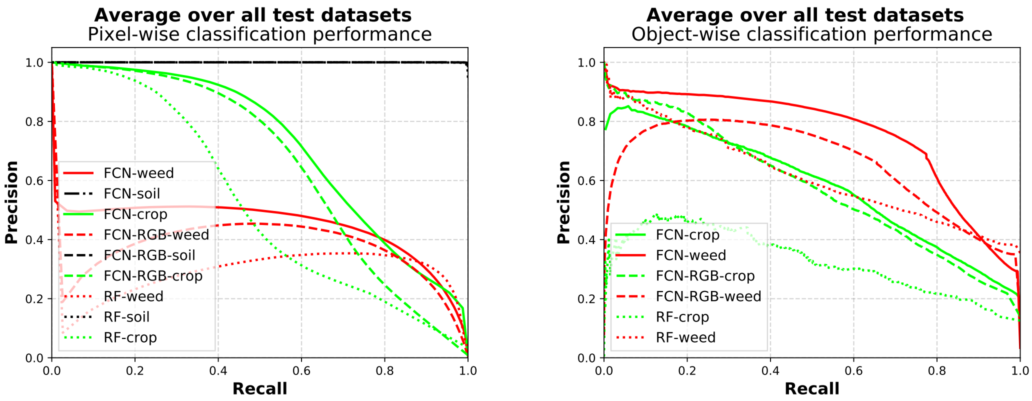


Figure 6.10: Precision-recall curves for the pixel-wise classification performance of the FCN, FCN-RGB, and RF-CAS approach. The precision-recall curve is computed using all test images from all test datasets. Thus, it reflects the generalization capabilities to new and unseen field environments.

Figure 6.10 depicts the precision-recall curves for the pixel-wise and object-wise crop-weed classification performance computed under the consideration of the entire set of test images coming from all test datasets, i.e., BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16. This performance reflects the average performance across all datasets and thus the generalization capabilities to new and unseen field environments. We observe an achieved average recall for weeds of around 85 % by the FCN, 77 % by the FCN-RGB, and 78 % with the RF-CAS approach. Thus, none of the tested approaches reaches the level of currently applied non-precision mechanical tools. Moreover, a robot equipped with these classifiers would accidentally remove a substantial amount of sugar beets in an autonomous weed control scenario by considering actual crop plants as weeds. The pixel-wise performance of the classifiers on the other test datasets is even lower. Neither the fully convolutional neural network approaches nor the random forest-based approach can provide suitable performance for autonomous field intervention in new field environments.

When comparing the fully convolutional neural network approaches and RF-CAS, it becomes clear that the fully convolutional neural network approaches outperform RF-CAS by around 10 % in terms of the average F1-score averaging across all datasets. This means that the fully convolutional neural network-based approaches serve better generalization capabilities to new and unseen fields compared to the random forest approach. Most errors of the RF-CAS approach are caused by the distinction of crop plants and weeds and less by the separation of vegetation and background. On the one hand, this means that fully convolutional neural networks can extract more general features for the classification task of crop and weed compared to the handcrafted features. On the other hand, the vegetation detection using NDVI generalizes well in new fields. These results are consistent with previous observations in Section 6.5.1 and Section 6.4.2.

Finally, we discuss the performance of FCN in comparison to the one obtained by FCN-RGB. In terms of the individual average F1-scores for the respective test datasets, the FCN provides a better crop-weed classification performance leading to an average gain of 3 % over its RGB-only variant. Differently than the performance under similar field conditions, the NIR information seems to be useful information for the fully convolutional neural network for the generalization capabilities in a new field.

Figure 6.11 illustrates qualitative classification results of the classifiers RF-CAS, FCN, and FCN-RGB under changing field conditions. The analysis of the qualitative result illustrates that all approaches substantially lack in performance when the visual appearance of the plants, weeds, and soil has notably changed between the training and deployment phase of the classifier.

6.5.2.1 Generalization Capabilities Under Training Data Diversity

In the previous experiment, we trained the classification models on the BONN-CW-16 dataset and deployed them for all other test datasets from different field environments. This experiment is designed to evaluate the effect on the model performance if we train

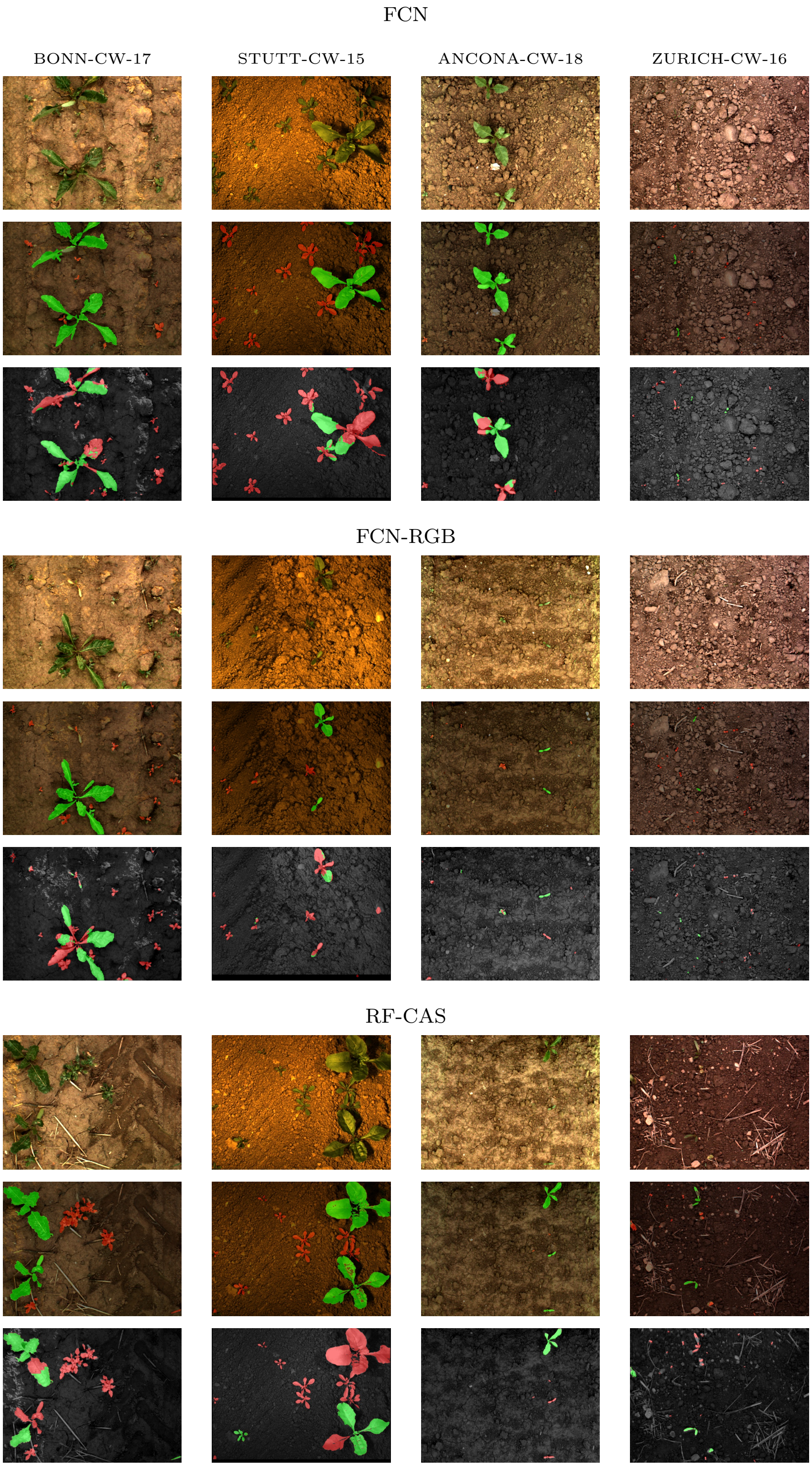


Figure 6.11: Qualitative results for the crop-weed classification performance of FCN, FCN-RGB, and RF-CAS under changing field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on NIR image. Crop plants (green) and weeds (red) represent the pixel-wise classification.

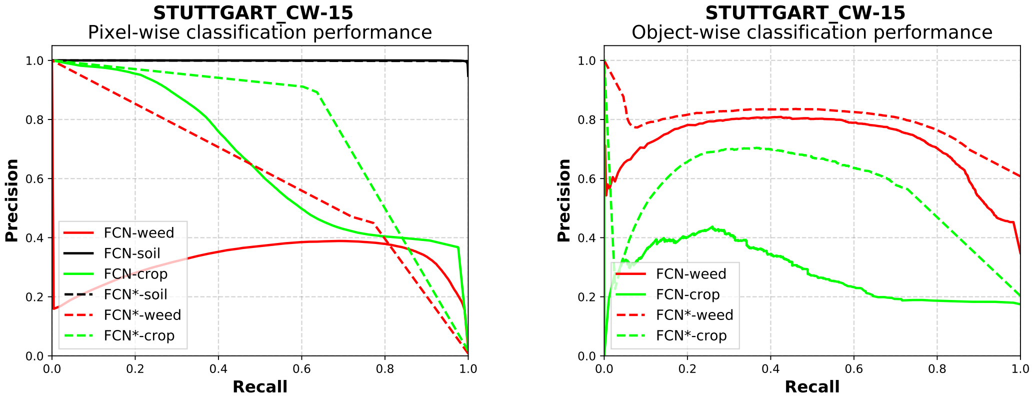


Figure 6.12: Precision-recall curves for the classification performance achieved by the FCN⋆ and FCN approach on the STUTT-CW-15 dataset. FCN⋆ achieves a better performance, especially in terms of the object-wise metric.

our model on a more extensive and even more diverse training dataset. The idea is to perform the training of the classifier already using data from different field environments containing crop plants, different weed species and at several growth stages, and different soil conditions. Furthermore, the data has been acquired under different illumination conditions. Thereby, we want to investigate whether the classifier is able to extract features that work more robust in changing field conditions. If this is the case, this experiment should lead to better performance on data coming to a further new field environment.

We train the FCN on 95 % of an aggregated dataset that involves the entire BONN-CW-16, BONN-CW-17, ANCONA-CW-18, and ZURICH-CW-16 datasets. We use the remaining 5 % portion as validation data to perform early stopping. Then, we deploy the classifier on the STUTT-CW-15 dataset. We expect that the classifier, which we call FCN⋆ in this experiment, provides better performance on the STUTT-CW-15 dataset compared to the performance obtained in the previous experiment when FCN was trained solely on the BONN-CW-16 dataset.

Figure 6.12 depicts the precision-recall curves for the pixel-wise and object-wise crop-weed classification performance achieved by the FCN⋆ and FCN approach on the STUTT-CW-15 dataset. The comparison shows that the classifier FCN⋆ performs better on the STUTT-CW-15 data than that of FCN. In other words, greater diversity in the training data can help to increase the generalization capabilities of the classification model. If we look at the results from an absolute performance point of view, even FCN⋆ cannot provide sufficient performance in a real-world application. Object-wise, it achieves a recall of 68 % at a precision of 60 % for the crop class. For weeds, it achieves a recall of 76 % at a precision of 79 % under labeling with the most likely class.

We also performed this experiment for other combinations of training and test data. We conclude that the generalization capabilities increase with the diversity of the training data, but the absolute performance does not reach a sufficient level for real-world applications such as robotic weed control. Nevertheless, these results show that it is worthwhile to build up a diverse training database to exploit the potential of diverse training datasets.

6.5.3 Conclusions for the Crop-Weed Classification Experiments

The results presented in this section lead to the following conclusions:

First, the fully convolutional neural network-based approaches provide better performance compared to the random forest-based approach in both cases, under similar and changing field conditions. We conclude that the learned features of the fully convolutional neural network approaches are more descriptive for the task, provide a better capacity, and generalize better to new fields compared to the handcrafted features used in the random forest.

Second, under similar field conditions, the fully convolutional neural network approaches do not necessarily rely on additional NIR information. Under changing field conditions, however, the NIR information aids the generalization capabilities.

Third, neither the fully convolutional neural network nor the random forest-based approaches provide suitable performance under changing field conditions. Thus, they are not capable of being applied in real-world applications when the classifier is trained once and then deployed in new and unseen field environments.

Fourth, for the FCN, a greater diversity of the training data induced by changing field conditions helps to learn better features to generalize to new field environments.

6.6 Vision-Based Crop-Weed Classification Exploiting Plant Arrangement

In the previous section, we showed that the performance of both the fully convolutional neural network-based and random forest-based crop-weed classification system is not reliable when deploying the classifiers in new and unseen field environments, i.e., in changing field conditions.

A focus of this thesis is the development of crop-weed classifiers that aim at bridging the lack of generalization capabilities to new field environments. To achieve more robust performance in these situations, we propose the approaches RF-GC in Section 4.4 and FCN-SEQ in Section 5.4 exploiting that within a field of row crops, the plants share a similar lattice distance along the row, whereas weeds appear more randomly. We evaluate our approaches RF-GC, which is a random forest-based classification system, as well as FCN-SEQ and its RGB-only variant FCN-SEQ-RGB, which are fully convolutional neural network classification systems. Note that we do not take into account the RGB-only version of RF-GC for the same reason as described in Section 6.5, namely a too low performance for the ExG-based vegetation classification under changing field conditions.

Table 6.9: Pixel-wise performance gain under similar field conditions of the sequential approaches FCN-SEQ, FCN-SEQ-RGB, and RF-GC compared to their non-sequential approaches FCN, FCN-RGB, and RF-CAS. We report the average F1-scores.

	ALL-DATA-CW avg. F1	BONN-CW-16 avg. F1
FCN-SEQ - FCN	3.0	4.1
FCN-SEQ-RGB - FCN-RGB	3.7	2.6
RF-GC - RF-CAS	6.0	5.0

Our sequential approaches combine visual features along with additional geometric features that exploit the spatial patterns of plant locations resulting from the sowing process. Figure 5.9 illustrates a classification result obtained by our FCN-SEQ approach on the STUTT-CW-15 dataset. The classification model was solely trained on another dataset from a different field environment, i.e., BONN-CW-16.

We separate the experiments analogously to the previous section into a performance evaluation under similar field conditions and under changing field conditions. To explicitly evaluate the effectiveness of using the plant arrangement information as an additional feature for the crop-weed classification task, we compare the results of FCN-SEQ and RF-GC with the one obtained by the purely visual approaches FCN and RF-CAS. We expect a better performance under similar field conditions and especially under changing field conditions, as FCN-SEQ and RF-GC exploit an additional source of information with the geometric features.

In this section, we consider the same datasets as for the crop-weed classification described in Section 6.5. We keep a 5 % held-out portion of the respective training data sets as validation data to perform early stopping.

6.6.1 Performance Under Similar Field Conditions

This experiment is designed to evaluate the performance under similar field conditions. We train the classification models on a 70 % training portion of the BONN-CW-16 and ALL-DATA-CW datasets and test their performance on a 25 % held-out portion, respectively.

First, we evaluate the effect on the performance when considering the spatial arrangement of the plants. We expect that the performance increases when we use additional geometric features for the classification as they encode somewhat independent information to the purely visual clues. That should intuitively help to solve the task. Therefore, we analyze the achieved performance of our sequential approaches FCN-SEQ, FCN-SEQ-RGB, and RF-GC and compare it with the performance obtained by our non-sequential approaches FCN, FCN-SEQ, and RF-CAS.

Table 6.9 summarizes the difference in the obtained average F1-score for the pixel-wise classification performance. The biggest beneficiary of geometric features is the random forest. RF-GC gains 6 % on the ALL-DATA-CW and 5 % on the BONN-CW-16 datasets compared to RF-CAS. In absolute numbers, RF-GC achieves a recall of

>93 % at precision >88 % for the crop class on both datasets. The geometric classifier of RF-GC uses the features of the probabilistic plant arrangement model, according to Equation (4.22). These features compensate for wrongly classified plants and weeds through the visual classifier. Furthermore, RF-GC constructs new trees for the random forest during the classification. Here, it considers examples, where the visual classifier is uncertain, but the geometric classifier is confident for a particular prediction, see Table 4.2. Through the help of the geometric information, the visual classifier also becomes better over time. Our investigations of those examples show that these are mostly predicted crop keypoints or objects that are spatially far away from the estimated crop row. The results suggest that the RF-GC approach can exploit the relative arrangement of the plants through the use of an additional geometric classifier exploiting the features of the probabilistic plant arrangement model.

For weeds, however, the recall obtained by RF-GC is 14 % better on BONN-CW-16 compared to the ALL-DATA-CW dataset. Here fully convolutional neural network approaches are more stable with a difference of around 3 %. The inner diversity in terms of weed types and field conditions of the ALL-DATA-CW data is higher than for the BONN-CW-16 data. We argue that the used handcrafted features of the random forest-based approach are not able to cover the high diversity in the ALL-DATA-CW dataset. Thus, the use of handcrafted features limits the capacity of the model to some degree.

Additionally, we evaluate the performance of the fully convolutional neural network approaches. The FCN-SEQ approach obtains an average F1-score of 92 % in terms of the pixel-wise classification performance on the ALL-DATA-CW datasets. This result reflects a gain of 3 % compared to the performance of its non-sequential variant, i.e., FCN. Also, for the approaches exploiting solely RGB as input, FCN-SEQ-RGB gains around 5 % compared to its non-sequential variant FCN-RGB. We observe a similar pattern for the performance comparison of the sequential and non-sequential approaches on the BONN-CW-16 dataset. These results show that the sequential, fully convolutional neural network approaches FCN-SEQ and FCN-SEQ-RGB can exploit the relative arrangement of the plants through analyzing image sequences of local field strips.

Next, we analyze the effect on the performance when using either RGB+NIR information or solely RGB information as input. Therefore, we compare the performance achieved by the FCN and FCN-RGB approach. Figure 6.13 depicts the performance of the fully convolutional neural network approaches in terms of precision-recall curves for every considered class on the ALL-DATA-CW dataset. On pixel-level and object-level, FCN-SEQ provides solid performance with higher precisions at higher recalls for both crop plants and weeds compared to FCN-RGB. Table 6.10 summarizes the performance for a labeling with the most likely class according to Equation (2.23). FCN-SEQ exploiting RGB+NIR archives a 2 % higher average F1-score on both datasets. We can conclude that, the exploitation of the additional NIR information also leads to a better performance in the case of the fully convolutional neural network approaches.

Another remarkable result is that FCN-SEQ-RGB, which uses RGB only, has even

Table 6.10: Pixel-wise crop-weed classification performance under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1
Training: ALL-DATA-CW (70%) Deployment: ALL-DATA-CW (25%)										
FCN-SEQ	92.0	87.6	97.6	95.8	93.0	98.7	80.3	70.0	94.2	99.9
FCN-SEQ-RGB	89.9	84.8	96.8	94.2	90.3	98.5	75.7	64.3	92.0	99.8
RF-GC	88.1	86.1	90.5	91.6	89.9	93.3	73.4	68.8	78.6	99.5
Training: BONN-CW-16 (70%) Deployment: BONN-CW-16 (25%)										
FCN-SEQ	93.8	89.8	98.8	97.0	94.5	99.6	84.6	75.0	96.9	99.9
FCN-SEQ-RGB	91.7	86.9	98.2	96.0	93.1	99.1	79.2	67.6	95.7	99.9
RF-GC	89.4	84.9	95.1	91.4	88.4	94.7	77.1	66.8	91.2	99.5

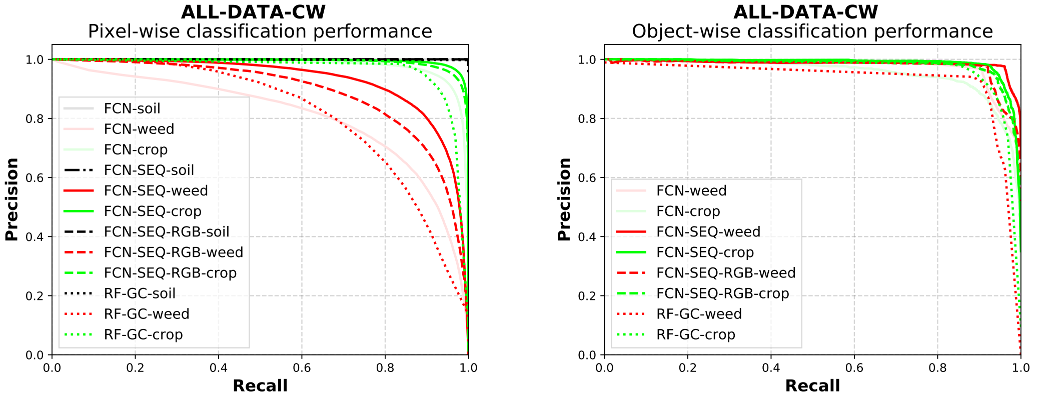


Figure 6.13: Precision-recall curves for the classification performance of the fully convolutional neural network approaches on the ALL-DATA-CW dataset. FCN has a slight advantage for the classification of weeds. For the object-wise performance the overall performance for both fully convolutional neural network approaches is comparable.

Table 6.11: Object-wise crop-weed classification performance under similar conditions. We report the class-wise and average F1-score (F1), precision (P) and recall (R) for a labeling with the most likely class according to Equation (2.23) in percent. The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Crop**			**Weed**		
	F1	P	R	F1	P	R	F1	P	R
Training: ALL-DATA-CW (70%) Deployment: ALL-DATA-CW (25%)									
FCN-SEQ	95.9	96.2	95.7	94.8	94.9	94.7	97.0	97.4	96.7
FCN-SEQ-RGB	94.6	95.1	94.2	93.7	92.7	94.8	95.4	97.4	93.5
RF-GC	78.2	80.2	79.5	74.0	87.6	64.0	82.4	72.7	95.0
Training: BONN-CW-16 (70%) Deployment: BONN-CW-16 (25%)									
FCN-SEQ	98.6	98.6	98.7	98.3	97.6	99.0	98.9	99.5	98.4
FCN-SEQ-RGB	97.9	98.2	97.6	97.7	97.3	98.2	98.1	99.1	97.1
RF-GC	87.2	86.0	88.6	88.0	85.9	90.2	86.5	86.0	87.0

slightly better performance on both data sets than FCN. Thus, the use of the spatio-temporal features extracted by the sequential module compensates for the need for the additional NIR information, at least under similar field conditions.

Finally, we analyze the best performing approach FCN-SEQ on object-level to obtain a performance estimate that is closer to the plant-level. Thereby, these results provide a better understanding of the expected performance in terms of the application of robotic weeding. Here, FCN-SEQ achieves an average F1-scores of 97 % for both datasets, ALL-DATA-CW and BONN-CW-16, with recalls of >95 % at a precision of >95 % for both the crop plants and weeds. Also the approaches FCN-SEQ and RF-GC achieve recalls of >95 % at a precision of >95 % for crops and >90 % at a precision of >90 % for weed. For the weed class, the FCN-SEQ approach performs slightly better than the RF-GC approach. Thus, all approaches classify the majority of plants and weeds correctly and obtain results that are suitable for the application in the field.

Figure 6.14 illustrates the qualitative results of the pixel-wise plant classification under similar field conditions. We show analyzed images from the experiment on the ALL-DATA-CW dataset. We sort the images according to the origin of the respective datasets BONN-CW-16, BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16. Visually, we can observe the slightly better performance for the fully convolutional neural network approaches compared to RF-GC. RF-GC sometimes produces false predictions for actual weeds that are located close to the crop row. Overall, all of the approaches provide reliable performance for the crop-weed classification task on all datasets.

We conclude that the fully convolutional neural network-based approaches classify the majority of objects correctly and provide substantially better performance than the random forest-based approach. The fully convolutional neural network features are more descriptive and provide a higher capacity to classify the data in terms of the crop-weed classification task under similar field conditions.

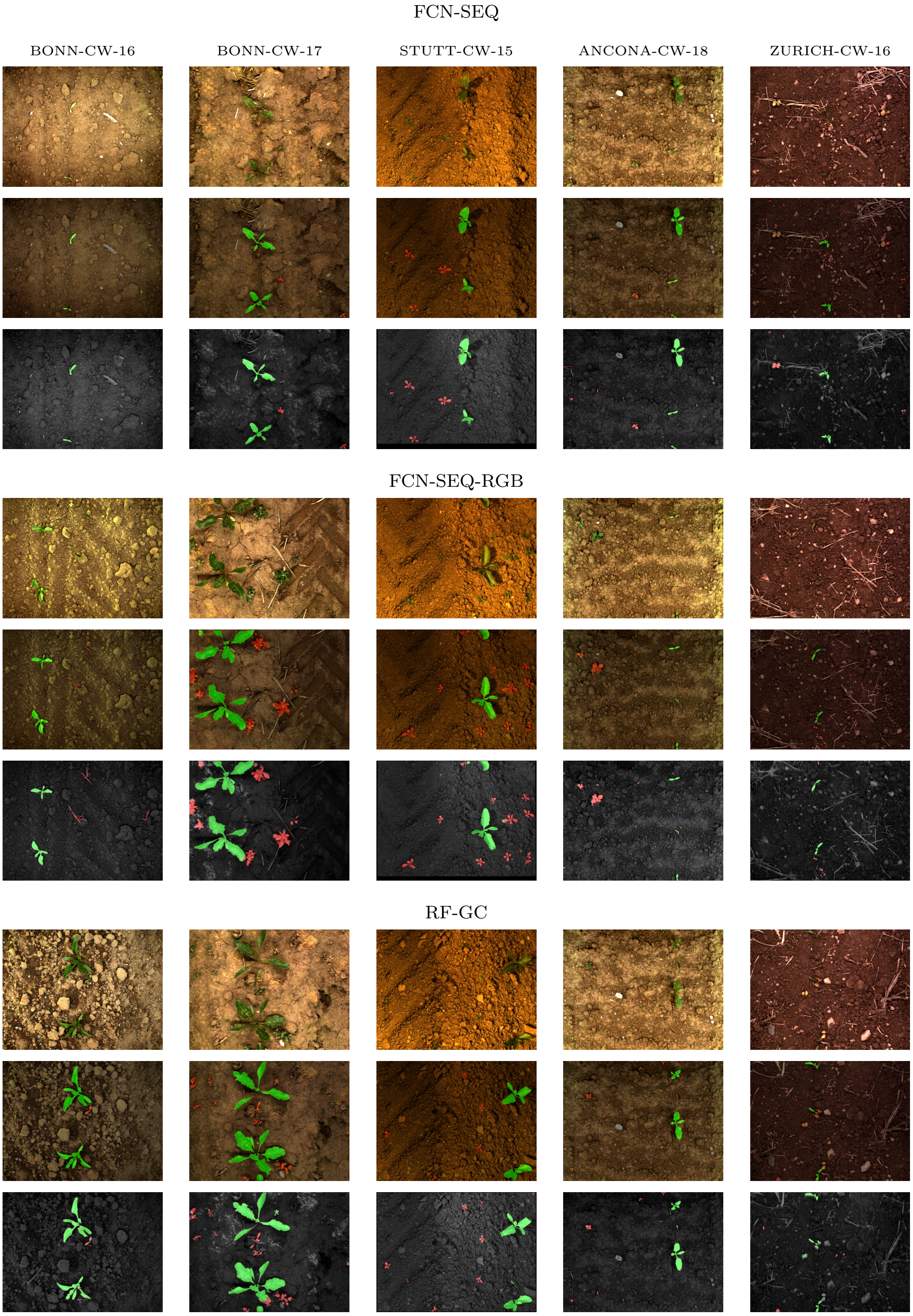


Figure 6.14: Qualitative results for the crop-weed classification performance of FCN-SEQ, FCN-SEQ-RGB, and RF-GC under similar field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on NIR image. Crop plants (green) and weeds (red) represent the pixel-wise classification.

Table 6.12: Pixel-wise performance gain under changing field conditions of the sequential approaches FCN-SEQ, FCN-SEQ-RGB, and RF-GC compared to their the non-sequential approaches FCN, FCN-RGB, and RF-CAS. We report the average F1-scores.

	BONN-CW-17	STUTT-CW-15	ANCONA-CW-18	ZURICH-CW-16
	avg. F1	avg. F1	avg. F1	avg. F1
FCN-SEQ - FCN	4.4	18.4	16.4	8.0
FCN-SEQ-RGB - FCN-RGB	3.1	16.6	11.3	1.7
RF-GC - RF-CAS	-10.0	-8.8	-9.2	-6.7

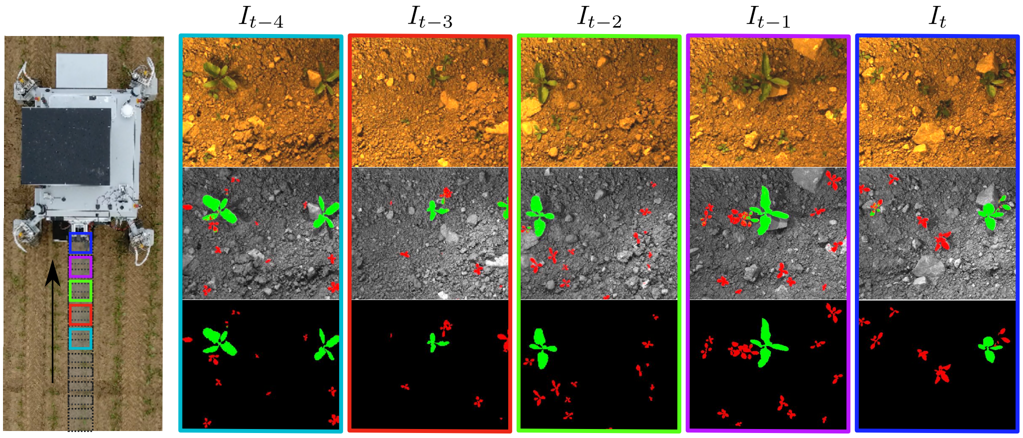


Figure 6.15: BoniRob acquiring images while driving along the crop row. We present exemplary prediction of crop plants and weeds for an image sequence of the STUTT-CW-15 data, where the classification model has been trained on the BONN-CW-16 dataset. The top row shows RGB images, the middle row shows the predicted label mask projected on the NIR image (crop in green, weed in red, background transparent), and the bottom row shows the ground truth for comparison. These results correspond to an average F1-Score of 87 %. For the entire test images of the STUTT-CW-15 dataset, we achieve around 84 %.

6.6.2 Performance Under Changing Field Conditions

We design the following experiment in this section to evaluate the performance under changing field conditions, i.e., to examine the generalization capabilities of our sequential approaches FCN-SEQ, FCN-SEQ-RGB, and RF-GC to new and unseen field environments. We train all classification models on a 95 % training portion of the BONN-CW-16 dataset and test them on the entire BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16 datasets.

First, we compare the performance of the sequential approaches with the non-sequential approaches. Thereby, we evaluate the effect of exploiting the information about the spatial arrangement on the generalization performance. Table 6.12 quantifies the performance gain achieved by our proposed FCN-SEQ, FCN-SEQ-RGB, and RF-GC approaches on every test dataset by showing the difference in the achieved performance regarding their corresponding non-sequential approaches. Table 6.13 summarizes the obtained pixel-wise crop-weed classification performance for all test datasets.

Table 6.13: Pixel-wise crop-weed classification performance under changing field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class according to Equation (2.23) in percent. The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-CW-16 Deployment: BONN-CW-17										
FCN-SEQ	82.9	79.5	89.5	83.3	86.9	80.0	65.8	52.2	88.9	99.5
FCN-SEQ-RGB	79.5	79.3	82.5	77.5	87.9	69.3	61.4	50.4	78.5	99.7
RF-GC	58.8	58.3	59.7	35.6	37.3	34.0	41.4	38.0	45.5	99.5
Training: BONN-CW-16 Deployment: STUTT-CW-15										
FCN-SEQ	84.7	82.0	87.7	87.2	82.1	92.9	67.2	64.2	70.6	99.7
FCN-SEQ-RGB	81.2	77.7	86.3	83.8	83.6	84.1	60.0	49.9	75.2	99.7
RF-GC	54.5	55.8	55.4	39.7	33.9	48.0	23.9	33.8	18.5	99.8
Training: BONN-CW-16 Deployment: ANCONA-CW-18										
FCN-SEQ	87.5	85.9	89.2	89.5	87.5	91.5	73.1	70.2	76.2	99.9
FCN-SEQ-RGB	84.9	85.1	85.1	85.1	80.7	89.9	69.9	74.7	65.6	99.9
RF-GC	59.1	57.9	60.9	40.9	35.6	48.1	36.7	38.6	35.0	99.6
Training: BONN-CW-16 Deployment: ZURICH-CW-16										
FCN-SEQ	70.8	62.8	85.3	51.3	42.0	66.0	61.4	46.6	90.1	99.7
FCN-SEQ-RGB	66.3	59.8	80.2	53.4	47.6	60.9	45.9	32.2	79.9	99.7
RF-GC	54.8	55.6	55.3	35.6	42.3	30.8	28.9	24.6	35.1	99.9
Average across all datasets										
FCN-SEQ	81.5	77.5	87.9	77.8	74.6	82.6	66.9	58.3	81.5	99.7
FCN-SEQ-RGB	78.0	75.5	83.5	75.0	75.0	76.1	59.3	51.8	74.8	99.8
RF-GC	56.8	56.9	57.8	38.0	37.3	40.2	32.7	33.8	33.5	99.7

Next, we analyze the achieved pixel-wise classification performance of the FCN-SEQ approach for the STUTT-CW-15 dataset. Regarding our investigations on the domain shift between the dataset in Section 3.2.1, this data set shows the most notable difference concerning the training data. FCN-SEQ achieves a recall of 95 % for the crop class and 71 % for the weed class leading to a performance gain of 18 % in average F1-score compared to the FCN approach. The same performance pattern holds for comparison of the RGB-only approaches FCN-SEQ-RGB and FCN-RGB.

Figure 6.16 depicts the precision-recall curves for the pixel-wise (left column) and object-wise (right column) crop-weed-classification performance under changing field conditions. At first glance, the curves illustrate the superior generalization capabilities of our proposed sequential approaches FCN-SEQ and FCN-SEQ-RGB. For all considered test datasets of different field environments, they provide superior generalization capabilities compared to the non-sequential FCN and FCN-RGB.

Except for the ZURICH-CW-16 dataset, the biggest performance gain is achieved for the crop class. Concerning the performance for the STUTT-CW-15 dataset, the sequential approaches obtain a performance boost, so that crop plants are classified with an F1 score of up to 85 % for the pixel-wise classification without re-training the classifier. Also, in the case of BONN-CW-17 and ANCONA-CW-18, the FCN-SEQ approach achieves recalls for the crop class of >90 % at a precision of 80 %. From this, we conclude that our proposed sequential module allows the extraction of useful features for better generalization capabilities to unseen field environments. On the ZURICH-CW-16 dataset, we observe mainly a notable pixel-wise performance boost for the weed class obtained by FCN-SEQ. We see two reasons for this. First, the quality of the plant arrangement is not as good as for the other datasets, see also our analysis in Section 6.8. Thus, the learned geometric information does not generalize well to ZURICH-CW-16. Second, the field in Zurich mostly contains very small plants, which reflects an additional challenge for the classifier to distinguish between the plants and weeds solely based on visual clues. Nevertheless, in terms of the object-based performance, we still see a notable improvement of around 10 % in recall and precision for the crop class for the sequential approaches. These results, however, suggest that a classifier deployed in a stage right after emergence still should be re-trained on data from the particular field environment.

Under consideration of the domain-shift between the datasets, see Section 3.2.1, we observe that the higher the domain-shift between the training and test domain, the higher the performance gain of the sequential fully convolutional neural network approaches. We argue that they can exploit the relative arrangement of the plants for extracting more descriptive features that are more robust to changing field conditions.

Figure 6.15 depicts the achieved performance of FCN-SEQ on the STUTT-CW-15 dataset, when the classifier has been trained solely on the BONN-CW-16 dataset. Aside from the fact that we perform the classification in a sequence-to-one fashion, we present the predictions across the whole sequence. This result supports the high recall obtained for the crop class. Furthermore, it can be seen that the crops and weed pixels are precisely separated from the soil, which indicates a high performance for the vegetation separation. The average F1-score for this sequence is about 87 %. For the entire test images of the STUTT-CW-15 dataset, we achieve around 84 %. We conclude that through the exploitation of spatio-temporal features, FCN-SEQ provides suitable results for autonomous weeding without the need to re-train the network for the field conditions in Stuttgart.

Next, we analyze the pixel-wise classification performance of the RF-GC approach. The results are contrary to the expectation that the use of field geometry leads to better performance. We determine that RF-GC does not provide usable results on the test data. Averaging the performance across all test datasets, the RF-GC approach "only" obtains a recall of 40 % for the crop class and 33 % for the weed class leading to an average F1-score of around 57 % (including the soil class).

We explore these initially unexpected results and find that the RF-GC performance drops due to the wrong initialization of the geometrical classifier. Initial wrong predictions of the visual classifier for the first few meters of the crop row can lead to the wrong estimate for the crop row. Since neither the visual nor the geometric classifier provide stable predictions, our strategies for combining both classifiers, which we describe in Table 4.2, fail. As a consequence of that, RF-GC produces a loop of misclassified examples that can also be fed as new training examples to the visual classifier from time to time. This not only prevents the visual classifier from adapting correctly to the current situation but also re-trains it with incorrect data. Figure 6.17 qualitatively illustrates the aforementioned problem. The wrongly classified crop plants indicate that the classification system converges to a wrong estimate of the crop row. These results indicate that RF-GC is not reliable for being deployed under changing field conditions if no training data is available from the targeted field environment to properly initialize its visual and geometric classifier.

Next, we analyze the generalization capabilities of the fully convolutional neural network approaches regarding their use of the additional NIR information. Therefore, we compare the classification performance between the approaches FCN-SEQ and FCN-SEQ-RGB. Figure 6.16 depicts the precision-recall curves for the pixel-wise (left) and object-wise (right) crop-weed-classification performance under changing field conditions. Each plot compares the performance between the FCN approaches when using RGB+NIR or solely RGB as input to the classification. The curves illustrate better generalization capabilities of the FCN-SEQ approach exploiting NIR information compared to its RGB-only variant FCN-SEQ-RGB. For a labeling with the most likely class according to Equation (2.23), see Table 6.13, FCN-SEQ obtains a better recall of around 6 % for crop and 6 % for weed when considering all test datasets. In terms of the precision for crop plants and weeds, the approach exploiting the additional NIR information gains around 2 % compared to its RGB-only variant. Thus, the additional NIR information aids the generalization capabilities of FCN-SEQ to new and unseen field environments.

Finally, we analyze the best-performing approach FCN-SEQ in regards to its applicability for real-world scenarios. Therefore, we analyze the performance on object-level to obtain a performance estimate that is closer to the plant-level. Here, FCN-SEQ

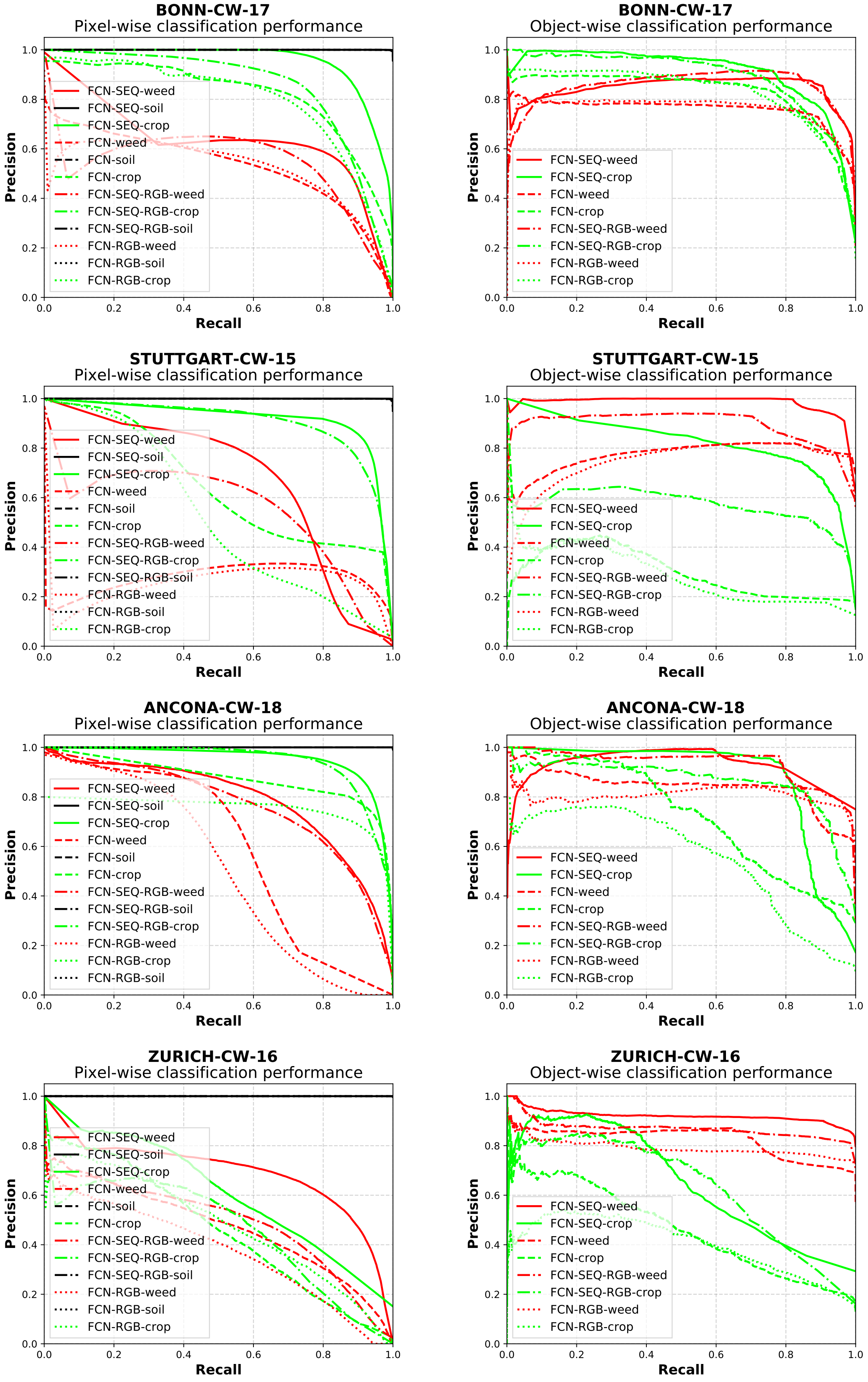


Figure 6.16: Precision-recall curves for the pixel-wise (left) and object-wise (right) crop-weed-classification performance under changing field conditions. The curves illustrate the superior generalization capabilities of FCN-SEQ.

achieves average F1-scores ranging from 81 %-85 % on the BONN-CW-17, STUTT-CW-15 and ANCONA-CW-18 datasets. Regarding the obtained recalls, this approach successfully detects 85 %-92 % of the actual crop objects and 77 %-91 % of the actual weed objects in the data. These results reflect a high performance considering that the classifier is not trained on samples coming from those particular fields. On the ZURICH-CW-16 data, however, FCN-SEQ achieves a rather low recall for the crop plants of around 55 %. Even if no other approach provides noticeably better performance on ZURICH-CW-16 when being trained on the BONN-CW-16 dataset, these results show that there are cases in which an adaptation of the classifier is indispensable to achieve a suitable performance for robotic weed control.

Figure 6.17 illustrates the qualitative results of the pixel-wise plant classification under different field conditions within the deployment phase of the classifiers RF-GC, FCN-SEQ, and FCN-SEQ-RGB. The analysis of the qualitative result illustrates that the fully convolutional neural network approaches provide good classification performance on the test datasets, whereas the random forest based approach fails due to a wrong initialization. Furthermore, it can be seen that the crops and weed pixels are precisely separated from the soil, which indicates a high performance for the vegetation separation.

Thus, this experiment shows the superior generalization capabilities of our proposed FCN-SEQ approach and demonstrates the positive impact on the performance when exploiting sequential data. The comparison of FCN-SEQ and FCN suggests that the additional arrangement information leads to a better classification performance and better generalization to new and previously unseen field environments.

6.6.3 Ablation Study of Key Architectural Design Choices for FCN-SEQ

In this experiment, we show the effect of the most central architectural design choices of our FCN-SEQ approach resulting in the largest gain and improvement of the performance under similar and changing field conditions.

We evaluate the performance of different architectural configurations by using the 70 % training data split of the BONN-CW-16 dataset for training and the 20 % test data split of BONN-CW-16 as well as the entire STUTT-CW-15, BONN-CW-17, ZURICH-CW-16, and ANCONA-CW-18 datasets for testing. We average the achieved performance for the latter test datasets and refer to it with the term ALLOTHER-CW. Thus, the results for the BONN-CW-16 datasets reflect the performance under similar field conditions, whereas the performance on ALLOTHER-CW data reflects the generalization performance to new and unseen field environments.

Table 6.14 reports the obtained object-wise average F1-scores. We start with a vanilla FCN corresponding to our encoder-decoder FCN approach without preprocessing its input. Then, we add our proposed preprocessing which helps to generalize to different fields. It minimizes the effect of different lighting conditions, as can be observed in Figure 4.2. We can furthermore improve the generalization capabilities by

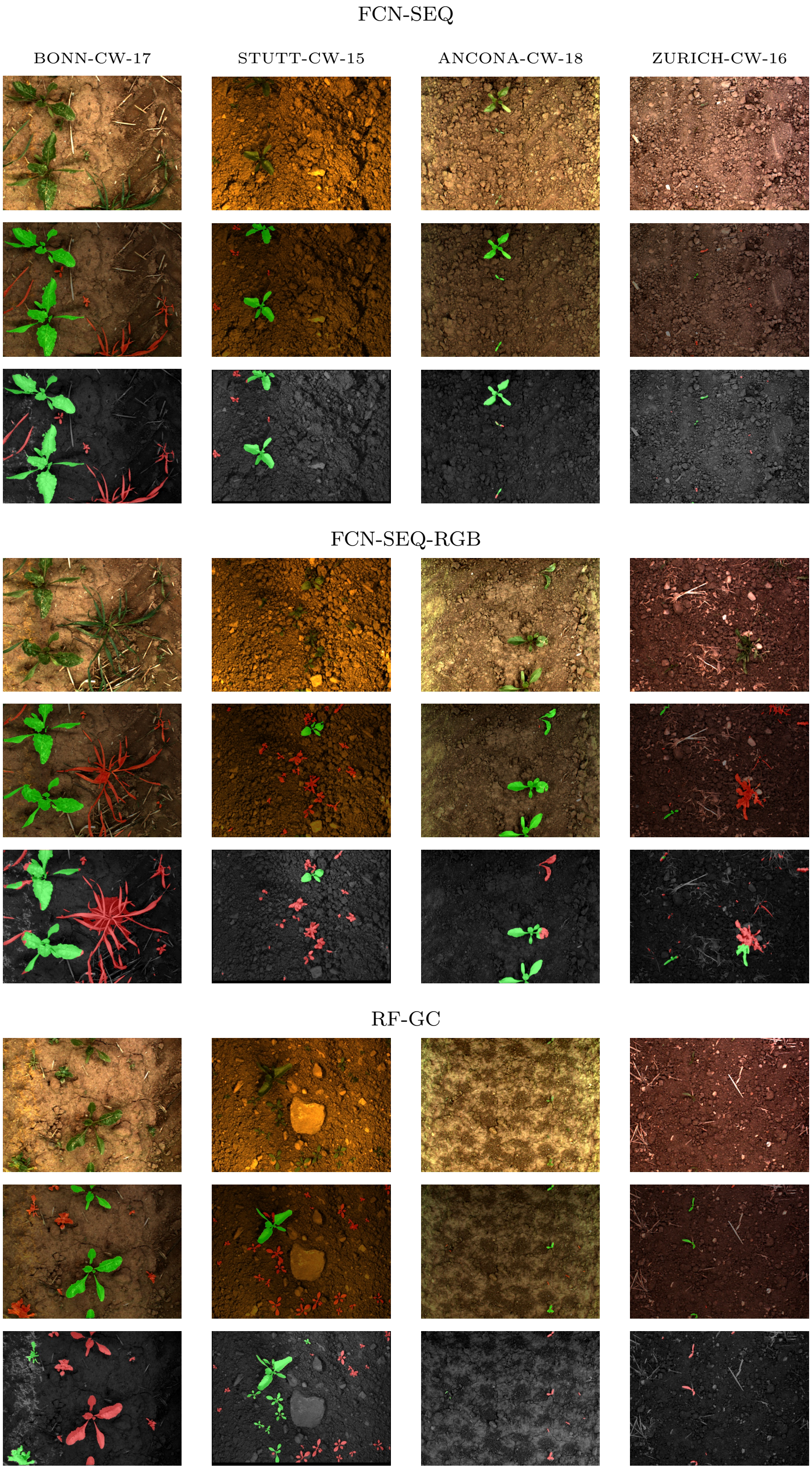


Figure 6.17: Qualitative results for the crop-weed classification performance of FCN-SEQ, FCN-SEQ-RGB, and RF-GC under changing field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on NIR image. Crop plants (green) and weeds (red) represent the pixel-wise classification.

Table 6.14: Ablation study for key components of the FCN-SEQ architecture. We report the object-wise average F1-score for the crop-weed classification.

Approach	BONN-CW-16	ALLOTHER-CW
All classifiers have been trained on 70 % of the BONN-CW-16 training data.		
Vanilla FCN	93.5	67.8
+ Preprocessing (FCN)	94.4	72.5
+ Sequential Module	95.6	77.8
+ Spatial Context (FCN-SEQ)	97.3	80.1
Gain	3.8	12.8

Table 6.15: Performance Evaluation of FCN-SEQ across different sequence lengths S.

S	BONN-CW-16	ALL-DATA-CW
	avg. F1	avg. F1
3	93.5	67.8
4	94.4	72.5
5	97.3	80.1
7	97.5	80.2
10	95.8	78.1

using the sequential module introduced in Section 5.4.2. Adding then more spatial context to the features through increasing the receptive field of the sequential module by using bigger kernels and dilated convolutions further improves the performance. The latter configuration corresponds to our proposed FCN-SEQ approach. In total we gain around 4 % performance under similar field conditions and 13 % performance under changing field conditions.

The high performance of all tested configurations for the held-out BONN-CW-16 dataset indicates that fully convolutional neural networks generally obtain a stable and high performance under a comparably low diversity in the data distribution. We conclude that preprocessing the input and the exploitation of the repetitive pattern given by the plant arrangement helps to improve the generalization capabilities of our FCN-SEQ plant classification system.

Finally, we evaluate the sequence length S, i.e., the number of images that form a sequence for the analysis. We examine the performance under the following sequence lengths $S \in \{3, 4, 5, 7, 10\}$. The results in Table 6.15 show that a sequence length of five or more produces the best results, and saturates for values larger than $S = 5$. This sequence length already provides enough information about the spatial distribution of the plants. In our setup, the length corresponds to a length of about 150 cm in the object space along the row, so we use $S = 5$, since a large valuer for S has a direct negative effect on the memory requirement and the runtime of the model. Moreover, during training time, we can train the model with $S = 10$ using only a batch size $B = 1$, as larger batch sizes lead to a too large memory consumption of the model on the GPU.

6.6.4 Simulation Experiments for Learning the Spatial Crop-Weed Arrangement

The main objective of the experimental design in this section is to evaluate our geometric approaches FCN-SEQ and RF-GC regarding their capabilities to extract information about the spatial plant arrangement to exploit this information for the crop-weed classification task. Due to a limited footprint of the UGVs camera system, see Section 3.1.1, a single image captured by the field robot does not cover a "large enough" area of the field to infer information about the spatial arrangement of the plants.

Figure 6.15 illustrates a sequence of images acquired while the robot traverses the field. Crop plants (green) grow in a row structure, sharing a similar spacing, whereas the weeds (red) are randomly located in the scene. In this experiment, we show that incorporating the sequential module in the FCN-SEQ approach described in Section 5.4.2 and the plant arrangement model in the RF-GC approach described in Section 4.4 enable the classifiers to identify the crop plants and weeds by *solely* considering their relative locations in the field.

To demonstrate this, we create synthetic images as input to the classification systems that provide only the signal of the plant arrangement as the potential information to distinguish the crop plants and weeds. We render simulated fields as depicted in Figure 6.18, encoding the locations of crops and weeds as uniformly sized circles. From these data, we extract a UGV-like acquisition setup of the camera and its motion in space to imitate the image sequences as they would have been acquired by the BoniRob field robot. Figure 6.19 illustrates four examples of simulated image sequences. We model the classes crop plants (green), weeds (red), and intra-row weeds (blue) in our simulator. We explicitly consider intra-row weeds, which are located close to the crop row, as they represent a special challenge for weeding tasks in precision-farming applications.

Figure 6.18 depicts a simulated crop field in terms of the plant, weed, and intra-row weed locations. We use this data as a basis to extract UGV-like image sequences. Our simulation allows us to model different properties for the arrangement of plants and weed pressure. For practical reasons, we constrain the parameters to lie within certain interval relevant for the applications, i.e., 10 cm-30 cm intra-row distance for crop plants, 30 cm-60 cm inter-row distance for crop rows, 0 %-500 % weed pressure considering the number of crop plants, and 0.5 cm^2-8.0 cm^2 plant size. In addition to that, we perpetuate the crop locations by Gaussian noise ($\mu = 0$ and $\sigma = 0.1\,\text{cm}$-$3.0\,\text{cm}$) in direction of the crop row and cross to it.

To imitate the acquisition setup of the BoniRob, we model the camera's motion with $0.2\frac{m}{s}$-$3.0\frac{m}{s}$ and the frame rate with 1 Hz-25 Hz. For the motion along the row, we also consider slight variations of the steering angle. Figure 6.19 depicts exemplary image sequences which are created under different properties. We use them for the training and testing of the classification systems. The first two columns show the respective input and output for training the classification system. Column 3-5 illustrate sequences build by different sets of properties. The input to the system are binary masks as shown

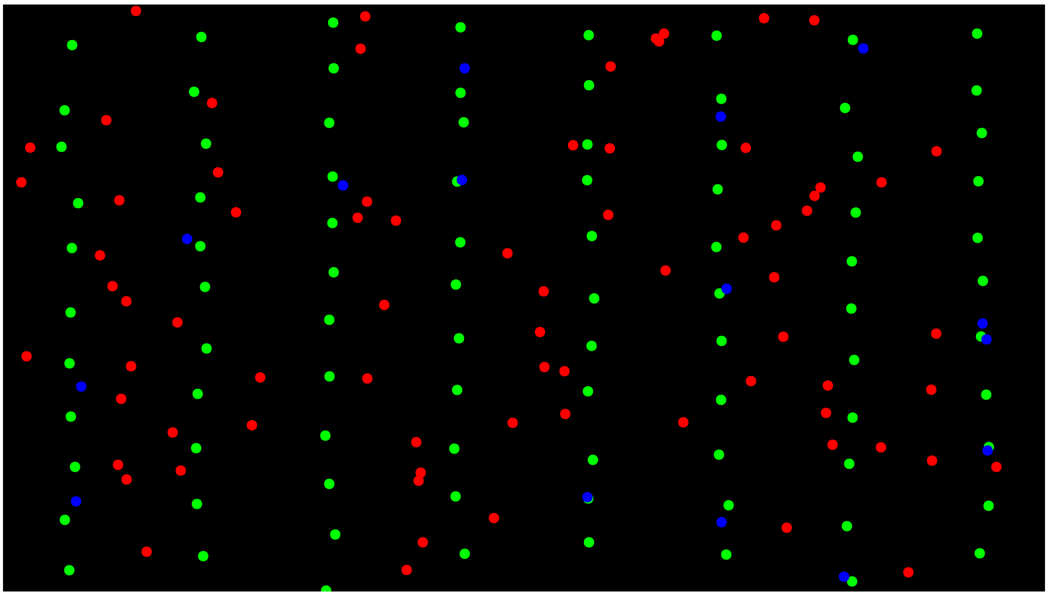


Figure 6.18: The plant simulator creates typical crop field situation modeling the locations of crop plants (green), weeds (red), and intra-row weeds (blue). Here: 50 cm±2 cm inter-row distance, 20 cm±3 cm intra-row distance, 4 cm^2 blob size imitating the plants, and a weed pressure of 200 %.

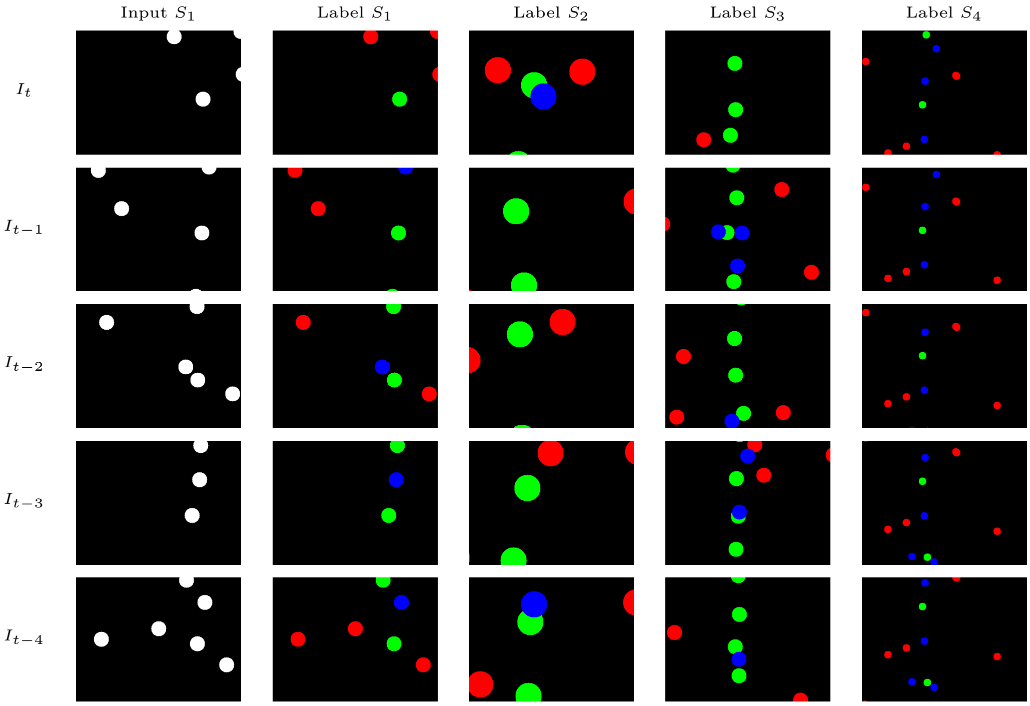


Figure 6.19: Four simulated image sequences. From left to right. first column: binary input data containing circles that represent different plants and weeds. second column: corresponding ground truth data. Crop plants (green), weeds (red) and intra-row weeds (blue). The blob size and shape is uniform within each sequence. Third column: sequence with bigger plant size. Fourth column: sequence with a lower intra-row-distance. Fifth column: sequence obtained by a higher frame rate of the camera.

Table 6.16: Object-wise classification performance on simulated images sequences. We report the class-wise and average F1-scores (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Crop**			**Weed**			**Intra-Weed**		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
Training: 5000 sequences Deployment: 2500 sequences												
FCN	39.1	37.1	41.7	57.7	57.3	58.2	54.9	49.0	62.4	4.7	4.9	4.5
FCN-SEQ	92.1	92.3	92.0	96.2	95.3	97.2	95.9	94.3	97.5	84.3	87.4	81.0
RF-CAS	40.1	38.8	42.1	53.4	47.9	60.4	56.6	55.9	57.3	10.3	12.7	8.7
RF-GC	65.4	59.4	72.7	77.9	69.3	89.0	81.9	74.4	91.2	36.2	34.6	38.0
Training: 5000 sequences Deployment: 2500 sequences Variance of intra-row distance restricted to 20 ± 3 cm												
RF-GC *	87.8	87.4	88.4	91.7	89.4	94.2	93.0	90.5	95.6	78.7	82.3	75.0

in column 1.

We hypothesize that an algorithm that solves the classification problem concerning the simulated image sequences described above must be able to learn features that describe the arrangement of crops and weeds. Note that within a given sequence, we model plants only by blobs of uniform size. Thus, we do not provide any spectral nor shape information as a potential input signal for distinguishing objects in the scene and thus for the classification task. One way to solve the problem, however, is to analyze image sequences together and to use the spatial arrangement of the objects.

We train all approaches on 5,000 simulated sequences and report the performance on another 2,500 sequences not used during training. We compare the performance of FCN-STEM and RF-GC exploiting visual and geometric features with the one obtained by FCN and RF-CAS, using solely visual features. Table 6.16 summarizes the obtained object-wise performance of the tested approaches. The results convey the superior performance of FCN-SEQ. It achieves an average F1-score of around 92 % across all classes. For the crop and the weed class, we can report a recall of 98 %. For the intra-weed class, we achieve an F1-score of 83 %, which indicates that FCN-SEQ is also able to exploit the intra-row distance between the crops along the crop row for the classification. We conclude that FCN-SEQ can exploit the sequential information to extract the pattern of the crop arrangement for the classification task.

Analyzing the results of RF-GC, it becomes visible that the performance does not reach the level of the FCN-SEQ approach. RF-GC achieves high recall for crop and weed in the order of 90 %, but a recall of 38 % for intra-row weeds. This result reflects the limited capacity of the plant arrangement model to encode different spatial arrangements induced by varying intra-row distances. Different intra-row distances between the plants in the training data lead to a probability mass $p(\boldsymbol{d} \mid \omega_{\mathsf{c}})$ according to Equation (4.19), that is mostly distributed along the crop row axis, see also Figure 4.13. Thus, it cannot properly distinguish the vegetation, especially in the intra-row space.

To prove this statement, we further investigated the performance of the RF-GC approach. We trained and deployed it only on data of an intra-row distance of 20 cm±3 cm. We refer to this experiment with RF-GC *. Here RF-GC * achieves results comparable to the FCN-SEQ approach with an average F1-score of 88 % caused by the higher recall for intra-row-weeds of around 75 %. In contrast, the achieved average F1-score of around 40 % for the baseline models indicates that FCN and RF-CAS are unable to accurately identify the crop plants and weeds as they cannot exploit the geometric signal.

6.6.5 Conclusions for the Crop-Weed Classification Classification Exploiting the Plant Arrangement Information

In total, the results convey the following outcomes:

First, the results support our claim of superior generalization capabilities for the fully convolutional neural network approaches exploiting the plant arrangement signal. Especially when the visual appearance of the image data notably differs between the training and test data, the spatio-temporal features extracted by the sequential module become key supporters for the performance. Nevertheless, exploiting the geometric features also help under similar field conditions.

Second, the RF-GC approach is not reliable for being deployed under changing field conditions if no training data is available from the targeted field environment to properly initialize its visual and geometric classifier. However, under similar field conditions, the system can be properly initialized. As a resume of that initialization, the geometric features lead to a notable boost in the performance as they can compensate for the limited capacity of the handcrafted visual features to some degree.

Third, the additional NIR information aids the generalization capabilities of FCN-SEQ compared to its RGB variant FCN-SEQ-RGB to new and changing field conditions.

Fourth, our simulation experiments demonstrate that our sequential FCN-SEQ approach can extract features that describe the relative arrangement of the plants and weeds. Furthermore, our RF-GC approach can exploit the geometry if the distribution of the arrangement keeps stable, which is the case for almost all crop row fields.

6.7 Joint Plant and Stem Detection for Species-Specific Treatments

We design the experiments in this section to analyze the quality of the vision-based plant classification pipeline for joint pixel-wise plant classification and stem detection enabling selective and plant-specific treatments, see Figure 6.20. The main objective of the tested approaches in this section is to simultaneously provide a pixel-wise classifica-

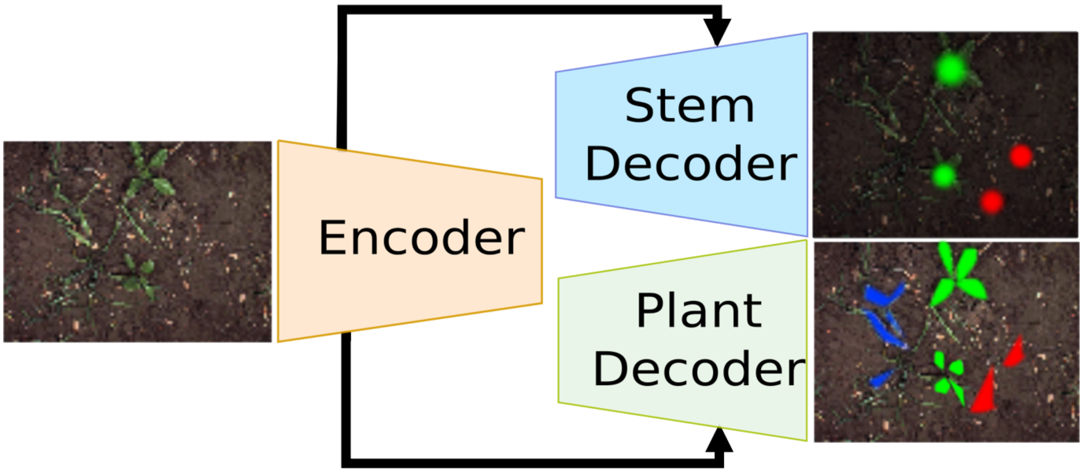


Figure 6.20: Concept for joint plant classification and stem detection. The main objective of FCN-STEM and FCN-SEQ-STEM is to provide two outputs simultaneously. First, a pixel-wise classification represented by the plant mask $\mathbf{I}_{\omega^{cdgs}}$ considering the classes $\omega^{cdgs} \in \{\omega_\mathsf{c}, \omega_d, \omega_g, \omega_s\}$ for crop, dicotyl weed, grass weed, and background (mostly soil). Second, the positions of the stems for dicotyl weeds and crop plants represented by the stem mask $\mathbf{I}_{\omega^{cds}}$ considering the classes $\omega^{cds} \in \{\omega_\mathsf{c}, \omega_d, \omega_s\}$ for crop stem, dicotyl weed stem, and no stem.

tion of the visual input into the classes crop, dicotyl weed, grass weed, and soil as well as providing the positions of the stems for dicotyl weeds and crop plants. The stem positions are a prerequisite in selective, high precision treatments, e.g., by mechanical stamping or by laser-based weeding. The provided pixel-wise label mask provides the area with more granulated approaches such as selective spraying.

We compare our proposed FCN-SEQ-STEM approach against its non-sequential version FCN-STEM. We make this comparison in order to understand the gain in performance due to the sequential module. FCN-STEM has the same architecture as FCN-SEQ-STEM without the sequential module. Both approaches can jointly estimate the plant stems and perform pixel-wise plant classification. In this case, the output of the proposed networks consists of two different label masks representing a probability distribution over the respective class labels. The first output is the plant label mask reflecting the pixel-wise classification of the crop plants, dicotyl weeds, and grass weeds, whereas the second output is the stem label mask segmenting regions that correspond to plant stems and dicotyl weed stems. Finally, we extract pixel-accurate stem positions from the stem label mask as described in Equation (5.3).

We also compare the performance of FCN-SEQ-STEM and FCN-STEM against our pixel-wise plant classification approaches FCN and FCN-SEQ. Here, we want to evaluate if the additional information induced by the stem detection task through a parallel task-specific decoder helps to improve the classification of the crop plants and weeds. Finally, we report the performance for a single image encoder-decoder FCN solely designed for stem detection. We refer to this approach with STEM. STEM has the same architecture as our FCN approach but is trained to output the stem label mask instead of the plant label mask. Finally, we also evaluate its sequential version, so-called STEM-SEQ. We compare the performance of all these approaches and show that sharing the encoder for stem detection and plant classification enables the networks to learn more descriptive features. Note that in this section, we do not evaluate the RGB-only variants of the tested approaches. We do not evaluate the performance for a random forest-based stem detection, as the evaluation in our previous publication [82] suggests that no suitable performance is achievable in this case.

Analogous to the previous experiments, we evaluate the performance under similar field conditions and evaluate the generalization capabilities of the classifiers under changing field conditions. In these experiments, we use different datasets as for the crop-weed performance evaluation in Section 6.5 and Section 6.6 for two reasons. First, the grass weed class is underrepresented in the crop-weed datasets that are described in Section 3.2.1. Therefore, we additionally labeled other parts of the recorded data containing more examples of grass weeds. Second, the manual labeling of stem locations for every crop plant and every dicotyl weed represents an additional labeling effort and thus was not performed for all images in the crop-weed datasets. We perform the experiments on the crop-dicot-grass datasets that are described in Section 3.2.2.

In this section, we use the following UGV datasets, which we describe in Section 3.2.2: BONN-CDGS-16, STUTT-CDGS-15, ANCONA-CDGS-18, and ZURICH-CDGS-17. All these datasets are fully labeled in a pixel-wise manner considering the classes crop, dicotyl weed, grass weed, and soil and regarding the stem positions for crop and dicot. The BONN-CDGS-16 dataset represents our primary source of training data. It involves around 2,400 labeled images. We acquired every other dataset on a different field at a different point in time. For the performance evaluation under similar field conditions, we consider the BONN-CW-16 dataset and the ALL-DATA-CDGS dataset, which is an aggregation of all crop-dicot-grass datasets. Here, we train the models on a training portion and test them on a held-out test portion, respectively. For the performance evaluation under changing field conditions, we solely train the models on the BONN-CW-16 data and test them on the other dataset, respectively. For all experiments in this section, we use a 5 % split of the training datasets as validation data for the fully convolutional neural network approaches to perform early stopping.

6.7.1 Stem Detection Performance

First, we analyze the stem detection performance and show that our approaches can accurately detect the stem locations of crop plants and dicotyl weeds under similar and under changing field conditions.

We consider a predicted stem to be a positive detection if its Euclidean distance to the nearest unassigned ground truth stem is below a threshold $\theta = 10\,\text{mm}$. We choose this threshold by keeping in mind the size of the mechanical stamping tool of the BoniRob. For the evaluation of the spatial accuracy of the detection in object space, we compute the mean average distance (MAD) in millimeters, taking all true positives into account.

Table 6.17: Stem detection performance of our proposed approaches under similar field conditions. We report the class-wise and average F1-score (F1), precision (P) and recall (R) as well as the mean average distance (MAD) in millimeters that measure the spatial accuracy of the stem predictions. We report the results for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**				**Crop**				**Dicot**			
	F1	P	R	MAD	F1	P	R	MAD	F1	P	R	MAD
Training: ALL-DATA-CDGS (70%) - deployment: ALL-DATA-CDGS (25%)												
STEM	69.6	64.1	76.3	3.3	71.7	64.0	81.4	3.6	67.5	64.2	71.2	2.9
FCN-STEM	74.4	72.5	78.7	2.9	69.7	59.9	83.2	3.8	79.2	85.0	74.1	2.0
STEM-SEQ	86.1	92.8	80.3	2.9	89.2	94.5	84.4	3.1	83.0	91.1	76.2	2.7
FCN-SEQ-STEM	90.3	97.8	84.0	2.7	92.9	97.8	88.5	2.9	87.7	97.7	79.5	2.5
Training: BONN-CDGS-16 (70%) - deployment: BONN-CDGS-16 (25%)												
STEM	77.7	67.6	91.5	3.6	79.0	69.3	91.9	3.9	76.5	65.9	91.1	3.3
FCN-STEM	87.3	83.0	92.9	3.0	93.1	92.8	93.4	3.1	81.6	73.1	92.3	3.0
STEM-SEQ	92.3	91.3	93.6	2.6	94.3	95.7	92.9	2.6	90.4	86.9	94.2	2.6
FCN-SEQ-STEM	92.8	93.8	91.8	2.4	93.4	95.2	91.6	2.3	92.1	92.3	92.0	2.5
Average performance under similar conditions												
STEM	73.7	65.9	83.9	3.4	75.3	66.7	86.7	3.8	72.0	65.1	81.2	3.1
FCN-STEM	80.9	77.7	85.8	3.0	81.4	76.4	88.3	3.5	80.4	79.1	83.2	2.5
STEM-SEQ	89.2	92.1	86.9	2.8	91.7	95.1	88.7	2.9	86.7	89.0	85.2	2.7
FCN-SEQ-STEM	91.5	95.8	87.9	2.6	93.1	96.5	90.1	2.6	89.9	95.0	85.8	2.5

6.7.1.1 Performance Under Similar Field Conditions

We start by evaluating the performance under similar field conditions. Therefore, we train the classification models on a 70 % training portion of the BONN-CDGS-16 and ALL-DATA-CDGS datasets and deploy them on 25 % test portions, respectively. Table 6.17 summarizes the obtained stem detection performance under similar field conditions. We see that FCN-SEQ-STEM outperforms the competing approaches on both datasets BONN-CDGS-16 and ALL-DATA-CDGS in terms of the achieved average F1-score. A better precision mainly causes better F1-score for both, the crop-score for both crop plants and dicotyl weeds.

With an average F1-score of around 93 % on BONN-CDGS-16 and 90 % on ALL-DATA-CDGS, the FCN-SEQ-STEM approach detects most of the stems correctly. Also, the STEM-SEQ approach obtains a comparable performance with 93 % average F1-score on BONN-CDGS-16 and 86 % on ALL-DATA-CDGS. On average, the sequential approaches exploiting spatio-temporal features that encode the arrangement of the plants outperform the non-sequential approaches STEM and FCN-SEQ-STEM by around 14 %. The analysis of the precision and recall values reveals that the performance benefits mainly in terms of precision. Thus, the approaches FCN-SEQ-STEM and STEM-SEQ provide less false detections for the crop and dicotyl weed stems. These results indicate that additional sequential information aids stem detection performance.

Furthermore, the results show that the performance of stem detection is substantially better for those classification systems that use two task-specific decoders for the classification of plants and detection of stems in parallel. FCN-SEQ-STEM reaches a 2 % higher average F1-score than STEM-SEQ, and FCN-STEM reaches a 8 % higher average F1-score than STEM. Thus, this effect seems to be greater, the more diverse the data is. We conclude that sharing one encoder aids the performance of the task of stem detection, as the extracted features can profit from both tasks, the classification of the plants and stem regions.

Concerning the average MAD for stems, we report the best performance for the FCN-SEQ-STEM approach ranging from 2.7 mm on the BONN-CDGS-16, and 2.4 mm on the ALL-DATA-CDGS datasets. The worst MAD performance is obtained by the STEM approach with 3.9 mm on the BONN-CDGS-16 dataset for the crop class. In total, these results are sufficient for the precise mechanical treatments using the BoniRob, but also, for even more precise laser-based weeding applications. Moreover, we observe that the sequential information aids the MAD directly by exploiting the geometric signal in the data and indirectly by improving the recall and precision for stem detection. Concerning both datasets, the sequential approaches achieve a 0.5 mm more accurate MAD.

In Figure 6.21, we illustrate qualitative results of our proposed approaches FCN-STEM and FCN-SEQ-STEM on the ALL-DATA-CDGS datasets. We see that most of the stems both for crops and for dicotyl weeds are detected correctly. Moreover, the results show that our approach can detect very small dicotyl weeds, which are of a size of around 0.15 cm^2, and only represented by a few pixels in the image.

6.7.1.2 Performance Under Changing Field Conditions

Next, we evaluate the performance under changing field conditions. Therefore, we train the classification models on 95 % portion of the BONN-CW-16 dataset and test them on all other datasets considered in this section. Table 6.18 summarizes the obtained stem detection performance under changing field conditions. Analyzing the performance on the datasets STUTT-CDGS-15, ANCONA-CDGS-18, and ZURICH-CDGS-17, we report the highest average F1-scores of 77 % for STUTT-CDGS-15, 75 % for ANCONA-CDGS-18, and 83 % for ZURICH-CDGS-17 obtained by our proposed FCN-SEQ-STEM approach. The second-best performance is achieved by the other sequential approach STEM-SEQ that solely predicts the stems for crop plants and dycotyl-weeds. Even if the performance under changing field conditions does not reach the same high level as under similar conditions, these results indicate the superior generalization capabilities to new and unseen field environments of the approaches exploiting the local plant arrangement in the field. We report a noticeable gain in the generalization performance of around 11 % in terms of average F1-score caused by the exploitation of the spatio-temporal features extracted by the FCN-SEQ-STEM approach.

FCN-SEQ-STEM achieves an 8 % better average F1-score compared to STEM-SEQ when averaging across all test datasets. Thus, using a shared encoder with two task-

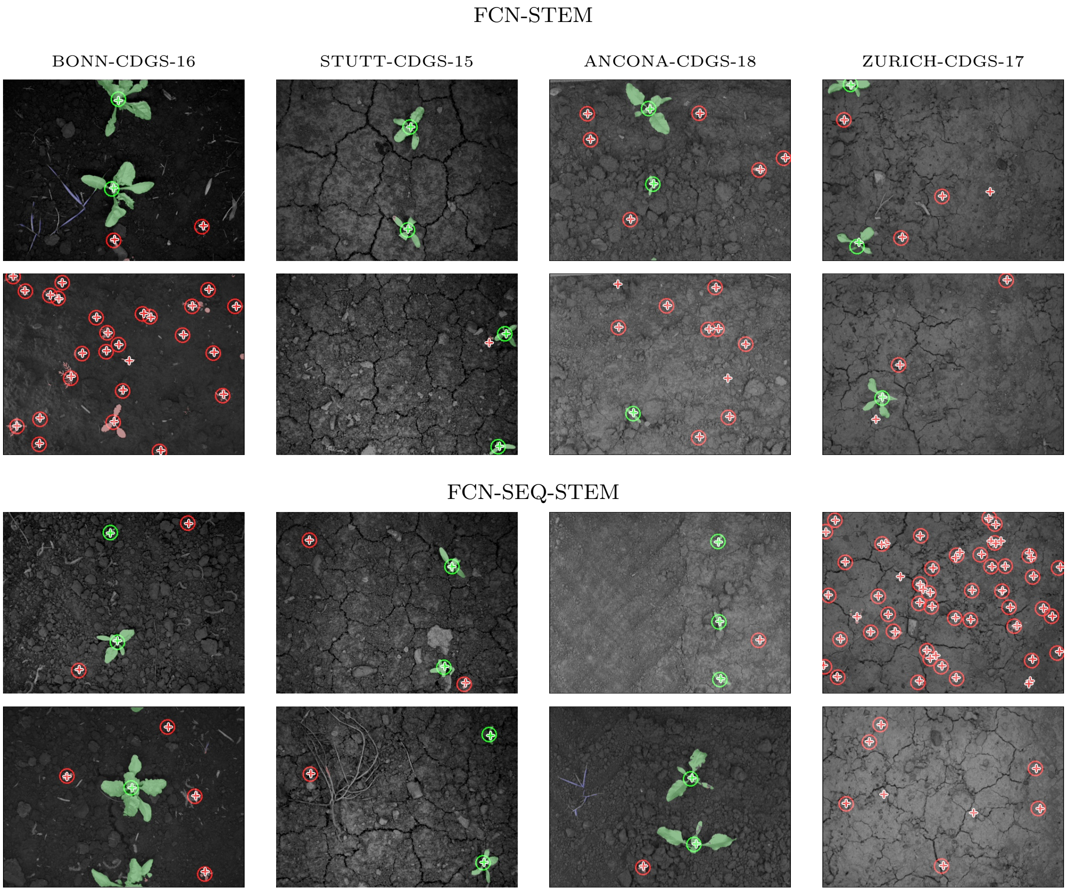


Figure 6.21: Qualitative results for the stem detection under similar field conditions obtained by our proposed approaches FCN-STEM and FCN-SEQ-STEM on the test datasets. We show two representative examples per dataset. We show an overlay of the NIR image with the prediction, where crop plants (green), dicotyl weeds (red), and grass weeds (blue) represent the pixel-wise classification. The predicted stems are illustrated by red (dicot) and green (dicot) cycles, whereas smaller filled circles illustrate the ground truth stems.

Table 6.18: Stem detection performance of our proposed approaches under changing field conditions. We report the class-wise and average F1-score (F1), precision (P) and recall (R) as well as the mean average distance (MAD) in millimeters that measure the spatial accuracy of the stem predictions. We report the results for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**				**Crop**				**Dicot**			
	F1	P	R	MAD	F1	P	R	MAD	F1	P	R	MAD
Training: BONN-CDGS-16- deployment: STUTT-CDGS-15												
STEM	42.4	50.5	49.4	3.6	51.8	78.3	38.7	4.3	33.0	22.7	60.1	2.9
FCN-STEM	58.9	57.0	63.0	2.9	69.8	73.6	66.4	3.1	48.1	40.3	59.5	2.6
STEM-SEQ	69.9	67.9	73.3	2.8	82.8	86.3	79.6	3.0	56.9	49.5	66.9	2.7
FCN-SEQ-STEM	77.4	73.4	82.4	2.7	83.6	82.3	84.9	2.8	71.3	64.4	79.8	2.6
Training: BONN-CDGS-16- deployment: ANCONA-CDGS-18												
STEM	47.4	49.4	51.9	3.7	53.8	67.3	44.8	4.2	41.1	31.5	59.0	3.2
FCN-STEM	59.3	65.8	54.2	3.2	69.0	79.9	60.7	3.3	49.6	51.7	47.7	3.0
STEM-SEQ	62.8	71.7	57.4	2.8	63.4	82.1	51.6	2.8	62.2	61.3	63.1	2.8
FCN-SEQ-STEM	74.8	75.7	75.1	2.8	77.7	84.9	71.7	2.8	71.9	66.4	78.4	2.9
Training: BONN-CDGS-16- deployment: ZURICH-CDGS-17												
STEM	60.1	54.5	76.5	3.7	52.8	37.5	89.2	4.7	67.4	71.5	63.8	2.7
FCN-STEM	76.5	80.5	74.8	3.0	77.2	71.5	83.8	3.4	75.8	89.5	65.8	2.5
STEM-SEQ	78.5	86.7	72.5	2.9	80.0	81.1	78.9	3.2	77.0	92.2	66.1	2.6
FCN-SEQ-STEM	83.1	86.2	81.0	2.7	80.7	77.5	84.2	2.9	85.4	94.9	77.7	2.4
Average performance under changing conditions												
STEM	50.0	51.5	59.3	3.7	52.8	61.0	57.6	4.4	47.2	41.9	61.0	2.9
FCN-STEM	64.9	67.8	64.0	3.0	72.0	75.0	70.3	3.3	57.8	60.5	57.7	2.7
STEM-SEQ	70.4	75.4	67.7	2.8	75.4	83.2	70.0	3.0	65.4	67.7	65.4	2.7
FCN-SEQ-STEM	78.4	78.4	79.5	2.7	80.7	81.6	80.3	2.8	76.2	75.2	78.6	2.6

specific heads leads to the learning of more descriptive features, also in terms of the generalization performance to new fields. This pattern is also visible when comparing the performance of the non-sequential approaches FCN-STEM and STEM.

In terms of MAD, the achieved performance for all tested approaches is on the same level as for the evaluation under similar field conditions. This means that the classifiers generally provide a high spatial precision when a stem is detected, but the precision and the hit rate (recall) of stems suffer under the changed conditions on the test datasets.

With regards to a weed control scenario, the robot would treat almost 80 % of the dicotyl weeds on the test datasets, assuming the actuator works error-free. Note that we obtain this performance without re-training the classifier with data coming from the targeted field environment. Further analyzing the precision for the dicot class, the majority of false detections are located on the soil. Thus, the robot would not accidentally eliminate a substantial number of crop plants. Comparing the performance under changing field conditions with the one obtained under similar field conditions, however, the recommended method in practice is still to adapt the model with data coming from the targeted field environment, as FCN-SEQ-STEM achieves a 12 % better average F1-score on the ALL-DATA-CDGS dataset.

In Figure 6.22, we illustrate qualitative results of our proposed approaches FCN-STEM and FCN-SEQ-STEM on the respective test datasets. We see that the FCN-STEM approach provides an unsuitable performance for a robotic weed control application, whereas FCN-SEQ-STEM still detects most of the plant and dicotyl weed stems correctly. Moreover, the results show that our approach can detect very small dicotyl weeds, which are of a size of around 0.15 cm^2, and are only represented by a few pixels in the image.

6.7.2 Pixel-Wise Crop-Dicot-Grass Classification Performance

In this section, we analyze the performance of the pixel-wise plant classification distinguishing crop plants, dicotyl weeds, grass weeds, and soil under similar and changing field conditions. Through the explicit consideration of dicotyl weeds and grass weeds, we enable the robot to treat these types of weed differently, e.g., selectively spraying the grass weeds and larger dicotyl weeds, whereas treating small dicotyl weeds mechanically. We show that our sequential approach FCN-SEQ-STEM provides state-of-the-art performance in terms of classification performance and generalization capabilities to new fields.

6.7.2.1 Performance Under Similar Field Conditions

We start by evaluating the performance under similar field conditions. Therefore, we train the classification models on a 70 % training portion of the BONN-CDGS-16 and ALL-DATA-CDGS datasets and deploy them on 25 % test portions, respec-

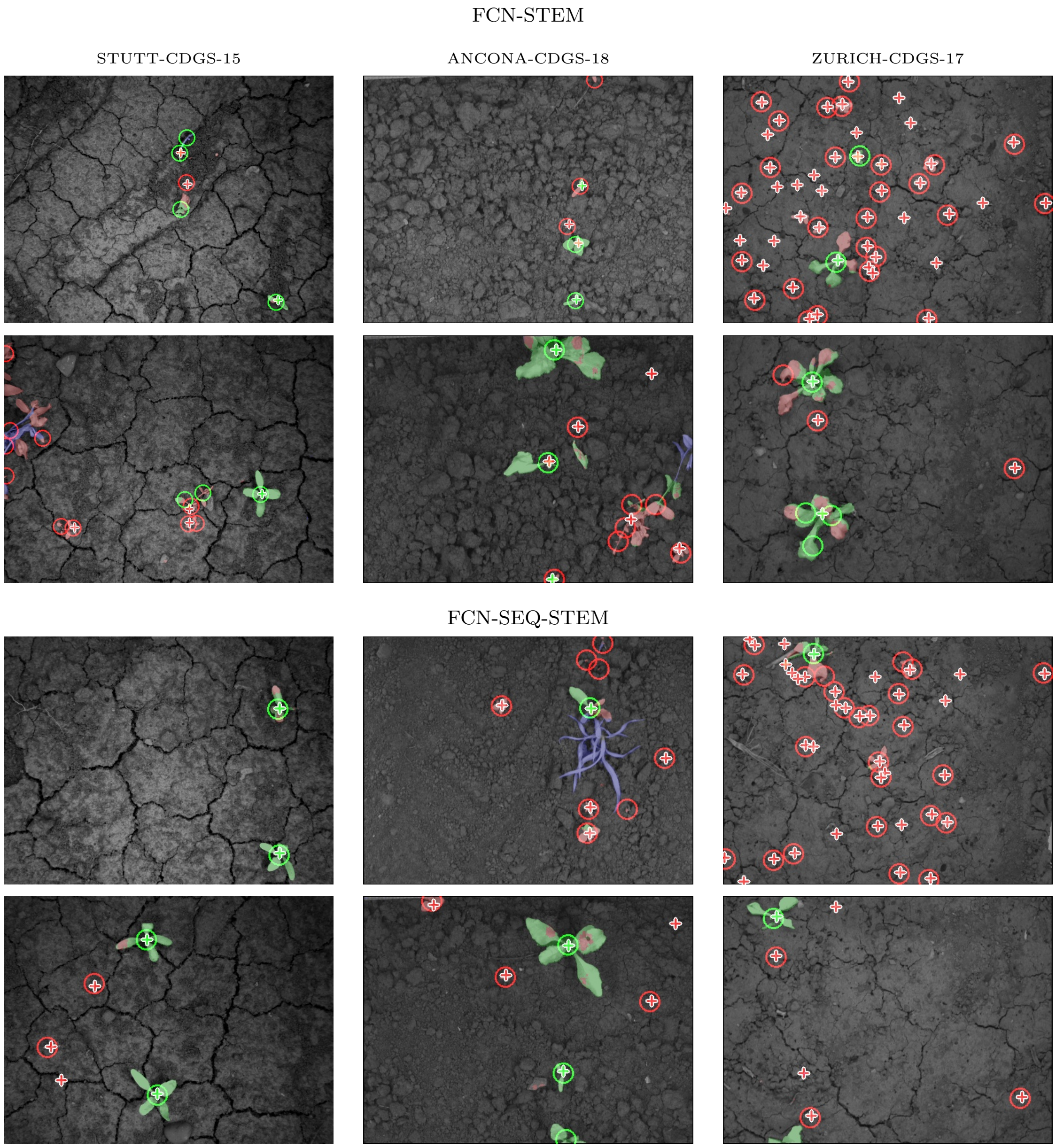


Figure 6.22: Qualitative results for the stem detection under changing field conditions obtained by our proposed approaches FCN-STEM and FCN-SEQ-STEM on the test datasets. We show two representative examples per dataset. We show an overlay of the NIR image with the prediction, where crop plants (green), dicotyl weeds (red), and grass weeds (blue) represent the pixel-wise classification. The predicted stems are illustrated by red (dicot) and green (dicot) cycles, whereas smaller filled circles illustrate the ground truth stems.

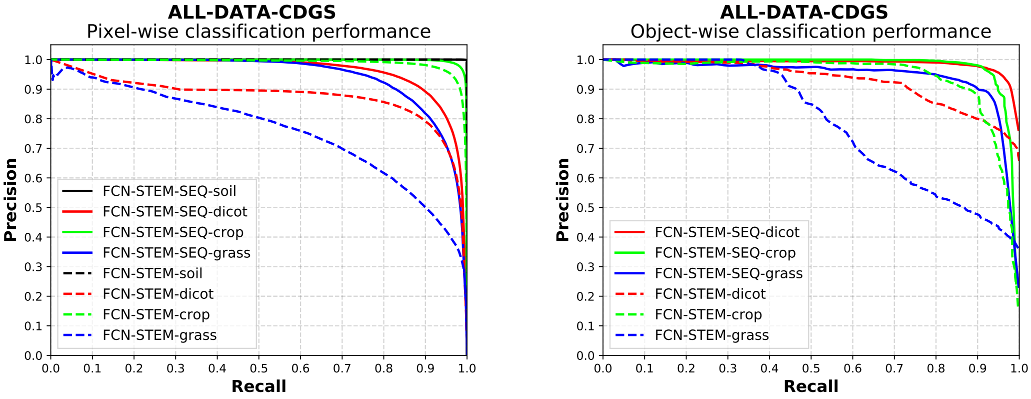


Figure 6.23: Precision-recall curves for the pixel-wise (left) and object-wise (right) crop-dicot-grass classification performance under similar field conditions.

Table 6.19: Pixel-wise crop-dicot-grass classification performance under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Dicot			Grass			Soil
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Training: ALL-DATA-CW (70%) Deployment: ALL-DATA-CW (25%)													
FCN	89.4	94.8	85.1	97.1	97.8	96.4	83.1	92.0	75.7	77.9	89.6	68.8	99.5
FCN-STEM	90.4	93.4	87.7	96.6	99.0	94.3	86.3	88.8	83.9	78.8	85.8	73.0	99.7
FCN-SEQ	90.4	94.7	86.7	97.0	98.5	95.5	84.1	91.8	77.6	80.7	88.7	74.0	99.7
FCN-SEQ-STEM	92.7	96.0	89.8	98.9	99.0	98.9	86.7	93.1	81.1	85.4	92.5	79.4	99.6
Training: BONN-CDGS-16 (70%) Deployment: BONN-CDGS-16 (25%)													
FCN	80.8	76.6	87.0	92.6	91.7	93.6	67.0	54.5	86.9	63.9	60.5	67.7	99.7
FCN-STEM	83.7	80.5	87.8	93.4	92.8	94.0	75.2	66.4	86.8	66.9	63.3	70.9	99.4
FCN-SEQ	85.7	82.4	89.8	93.8	93.1	94.5	80.1	70.4	92.9	69.4	66.7	72.2	99.6
FCN-SEQ-STEM	89.9	85.6	95.1	96.7	95.6	97.9	84.7	76.0	95.7	78.4	71.5	86.9	99.6
Average performance under similar conditions													
FCN	85.1	85.7	86.0	94.9	94.8	95.0	75.0	73.3	81.3	70.9	75.1	68.3	99.6
FCN-STEM	87.1	86.9	87.8	95.0	95.9	94.2	80.8	77.6	85.3	72.9	74.5	71.9	99.6
FCN-SEQ	88.0	88.6	88.3	95.4	95.8	95.0	82.1	81.1	85.3	75.0	77.7	73.1	99.7
FCN-SEQ-STEM	91.3	90.8	92.4	97.8	97.3	98.4	85.7	84.5	88.4	81.9	82.0	83.1	99.6

tively. Table 6.19 summarizes the obtained pixel-wise plant classification performance on both datasets. Overall, we find the same patterns in view on performance as for the stem detection in the previous experiments. First, the sequential approaches FCN-SEQ-STEM and FCN-SEQ exploiting spatio-temporal features perform better than the non-sequential ones. Second, the use of a shared encoder by two task-specific decoders in FCN-STEM and FCN-SEQ-STEM leads to a better pixel-wise classification performance compared to single-encoder-decoder networks FCN and FCN-SEQ.

The class-wise precision-recall plots in Figure 6.23 illustrate the achieved performance gain on the ALL-DATA-CDGS dataset when exploiting sequential input, i.e., considering the plant arrangement of plants in local field strips as also evaluated in Section 6.6. FCN-SEQ-STEM achieves a substantially higher recall at higher precision for both dicotyl weeds and grass weeds compared to FCN-STEM. In terms of labeling with the most likely class summarized in Table 6.19, FCN-SEQ-STEM obtains a higher precision of 6 % for the dicot and 13 % for the grass class compared to its non-sequential variant FCN-STEM. With a resulting average F1-score of around 93 %, FCN-SEQ-STEM classifies most of the pixels on the 25 % test portion correctly. Thus, the spatio-temporal features of the sequential approaches help to compensate for the confusion between the vegetation classes. In terms of the object-wise metric, FCN-SEQ-STEM achieves class-wise F1-scores >90 % for all classes.

Compared to the pixel-wise performance, the object-wise performance of FCN-SEQ-STEM is higher in terms of precision for the grass and dicot class. The discrepancy is mostly caused by predicted pixels in border regions of the vegetation objects. Even with pixel-wise recalls of 88 % for the dicot and 79 % for grass class, the system correctly recognizes the majority of vegetation objects with an average recall of 93 % at a precision of 92 %. Figure 6.24 qualitatively supports this statement for the FCN-SEQ-STEM approach. The visual inspection of the predictions and corresponding ground truth reveals that FCN-SEQ-STEM can properly classify the crop plants, even if they overlap with weeds in image-space. Moreover, it correctly identifies the grass weeds as well as large and tiny dicotyl weeds.

For FCN-STEM, however, we observe a notable amount of false predictions. The misclassification is caused by the confusion between the vegetation and soil as well as between the dicotyl weeds and grass weeds. This causes a lower overall performance of 88 % for the pixel-wise and 79 % for the object-wise performance in terms of average F1-score. We see two reasons for this performance loss. First, the confusion between the grass and dicotyl weeds leads to a decrease in recall and precision for both classes. Second, pixel-wisely, the weed classes occur less often in the data compared to crop and soil. Thus, their class-wise performance measures are more affected by false classifications.

We observe the same performance patterns by evaluation of the results on the BONN-CDGS-16 data. Thus, we conclude that exploiting the sequential data stream enables the networks to extract better features for distinguishing the considered classes. Concerning the grass class, however, all tested approaches obtain a lower performance on the BONN-CDGS-16 dataset compared to ALL-DATA-CDGS. The losses are explained by the low probability of occurrence for the grass class in the BONN-CDGS-16 data compared to the ANCONA-CDGS-18 that is part of the ALL-DATA-CDGS dataset, see Section 3.2.2. Furthermore, The grasses in the ANCONA-CDGS-18 have a larger spatial extent. Thus, the typical prediction errors in border regions of the grass weeds have less influence on the pixel-wise classification performance. This statement is also supported by the object-wise performance for grass, which is comparable for both datasets.

Next, we analyze the performance when using a shared encoder with two task-specific decoders. Therefore, we compare the relative performance between FCN and FCN-STEM as well as between FCN-SEQ and FCN-SEQ-STEM. In both cases, we see a performance boost for those networks that simultaneously classify the plants and stem regions. Averaging the performance across the BONN-CDGS-16 and ALL-DATA-CDGS datasets, the non-sequential approach FCN-STEM obtains a 6 % higher average F1-score than the single-encoder-decoder network FCN. For the sequential approaches, this performance gain is around 2 %. These results indicate that the stem detection task loss also leads to the extraction of more descriptive features for the plant classification. Intuitively, the stem information can help to classify grass weeds better. Furthermore, the stem locations can provide a more distinct signal of the plant arrangement, as this signal is not smudged by the spatial extent of the plants.

6.7.2.2 Performance Under Changing Field Conditions

Next, we evaluate the performance under changing field conditions. Therefore, we train the classification models on a 95 % training portion of the BONN-CDGS-16 dataset and deploy them on the entire considered test datasets. Table 6.20 summarizes the obtained pixel-wise classification performance under changing field conditions. First of all, we see that the overall performance is generally lower than in the evaluation under similar conditions. This is because the visual classifiers suffer from changes in the underlying intensity distribution between the test and training data. This observation is consistent with the crop-weed experiments from Section 6.5.2.

First, we compare our sequential approach FCN-SEQ-STEM with FCN-STEM to analyze the effect of the spatio-temporal features extracted from images sequences on the generalization capabilities to changing field conditions. On all three tested datasets, FCN-SEQ-STEM performs substantially better than FCN-STEM. This holds for both, pixel-wise as well as object-wise performance. On average, across all test datasets, it obtains a gain of around 12 % in terms of pixel-wise average F1-score. Figure 6.25 depicts the resulting precision-recall curves for the achieved performance on the STUTT-CDGS-15, ANCONA-CDGS-18, and ZURICH-CDGS-17 datasets. In all cases, we see a notable improvement in performance for FCN-SEQ-STEM. The crop class performance, in particular, benefits the most from the plant arrangement information. The resulting high recall for the plants means that the robot would not erroneously eliminate the crop plants because it considers them as weeds. In terms of performance for the STUTT-CW-15 dataset, we can even say that with the FCN-

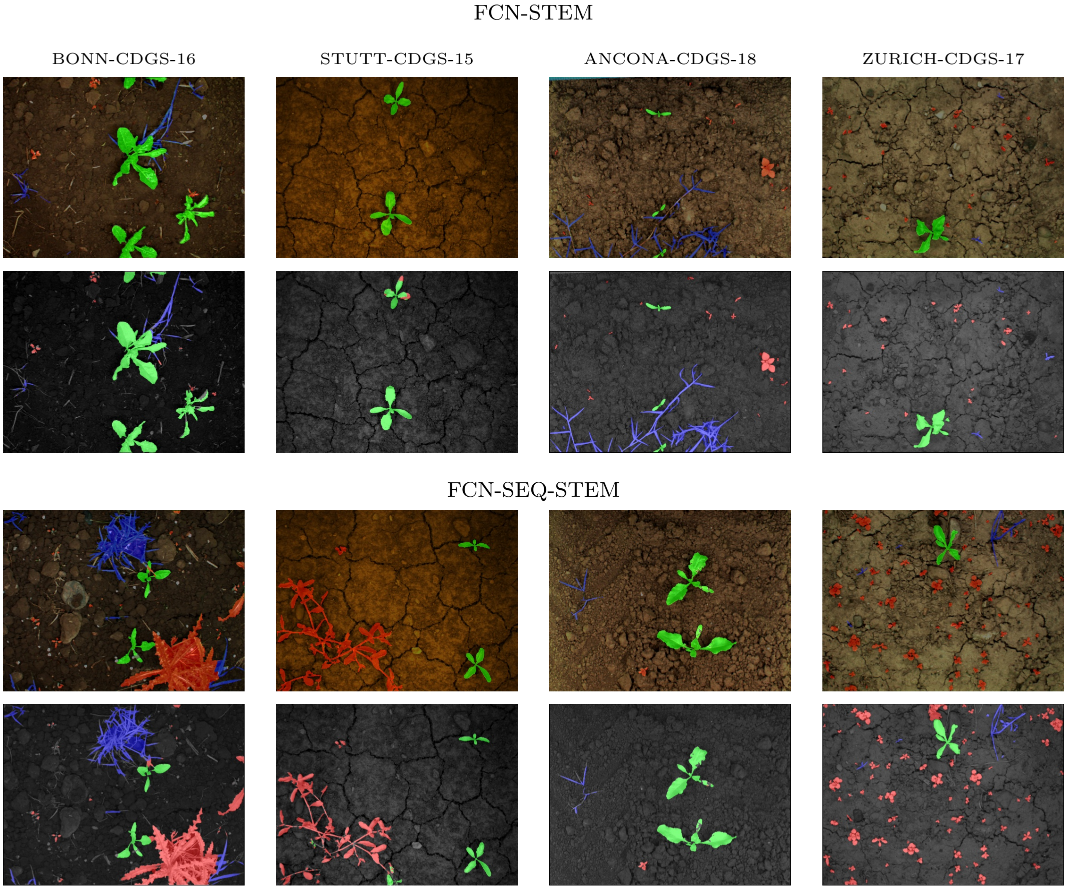


Figure 6.24: Qualitative results for the crop-dicot-grass classification performance of FCN-STEM and FCN-SEQ-STEM under similar field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on NIR image, where crop plants (green), dicotyl weeds (red), and grass weeds (blue) represent the pixel-wise classification.

Table 6.20: Pixel-wise crop-dicot-grass classification performance of our proposed approaches under changing field conditions. We report the class-wise and mean (m) F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Crop**			**Dicot**			**Grass**			**Soil**
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Training: ALL-DATA-CW (95%) Deployment: STUTT-CDGS-15													
FCN	56.2	68.1	57.3	35.0	78.9	22.5	33.8	25.6	49.6				99.7
FCN-STEM	58.0	69.7	58.7	38.6	81.8	25.3	35.8	27.5	51.2				99.7
FCN-SEQ	70.1	76.4	69.8	63.3	90.6	48.6	47.4	38.6	61.3				99.7
FCN-SEQ-STEM	77.6	73.1	83.3	81.6	74.5	90.1	51.6	45.2	60.1				99.6
Training: BONN-CDGS-16 (95%) Deployment: ANCONA-CDGS-18													
FCN	62.0	64.6	66.5	65.6	90.1	51.5	37.2	25.9	65.7	45.5	42.6	48.8	99.9
FCN-STEM	68.2	65.3	74.4	83.1	87.0	79.5	44.8	33.2	68.8	45.1	41.1	49.9	99.7
FCN-SEQ	76.3	73.6	81.4	86.8	85.4	88.2	51.1	39.6	72.0	67.6	69.7	65.6	99.8
FCN-SEQ-STEM	77.7	74.7	83.0	88.6	87.2	90.1	51.6	40.1	72.5	70.6	71.6	69.7	99.7
Training: BONN-CDGS-16 (95%) Deployment: ZURICH-CDGS-17													
FCN	54.6	61.1	52.8	52.4	74.0	40.5	52.6	45.9	61.5	13.8	25.0	9.5	99.6
FCN-STEM	65.1	66.3	64.8	71.5	72.1	70.9	60.3	56.1	65.2	28.5	37.1	23.2	99.9
FCN-SEQ	67.9	67.4	69.4	80.4	79.6	81.2	62.3	55.3	71.3	29.3	35.1	25.2	99.7
FCN-SEQ-STEM	71.8	72.9	71.0	83.4	81.4	85.6	71.2	72.9	69.6	32.7	37.5	29.0	99.9
Average performance under changing conditions													
FCN	57.6	64.6	58.9	51.0	81.0	38.2	41.2	32.5	58.9	29.6	33.8	29.2	99.6
FCN-STEM	63.8	67.1	66.0	64.4	80.3	58.6	47.0	38.9	61.7	36.8	39.1	36.6	99.6
FCN-SEQ	71.5	72.5	73.5	76.8	85.2	72.7	53.6	44.5	68.2	48.5	52.4	45.4	99.7
FCN-SEQ-STEM	75.7	73.6	79.1	84.5	81.0	88.6	58.2	52.7	67.4	51.7	54.6	49.4	99.7

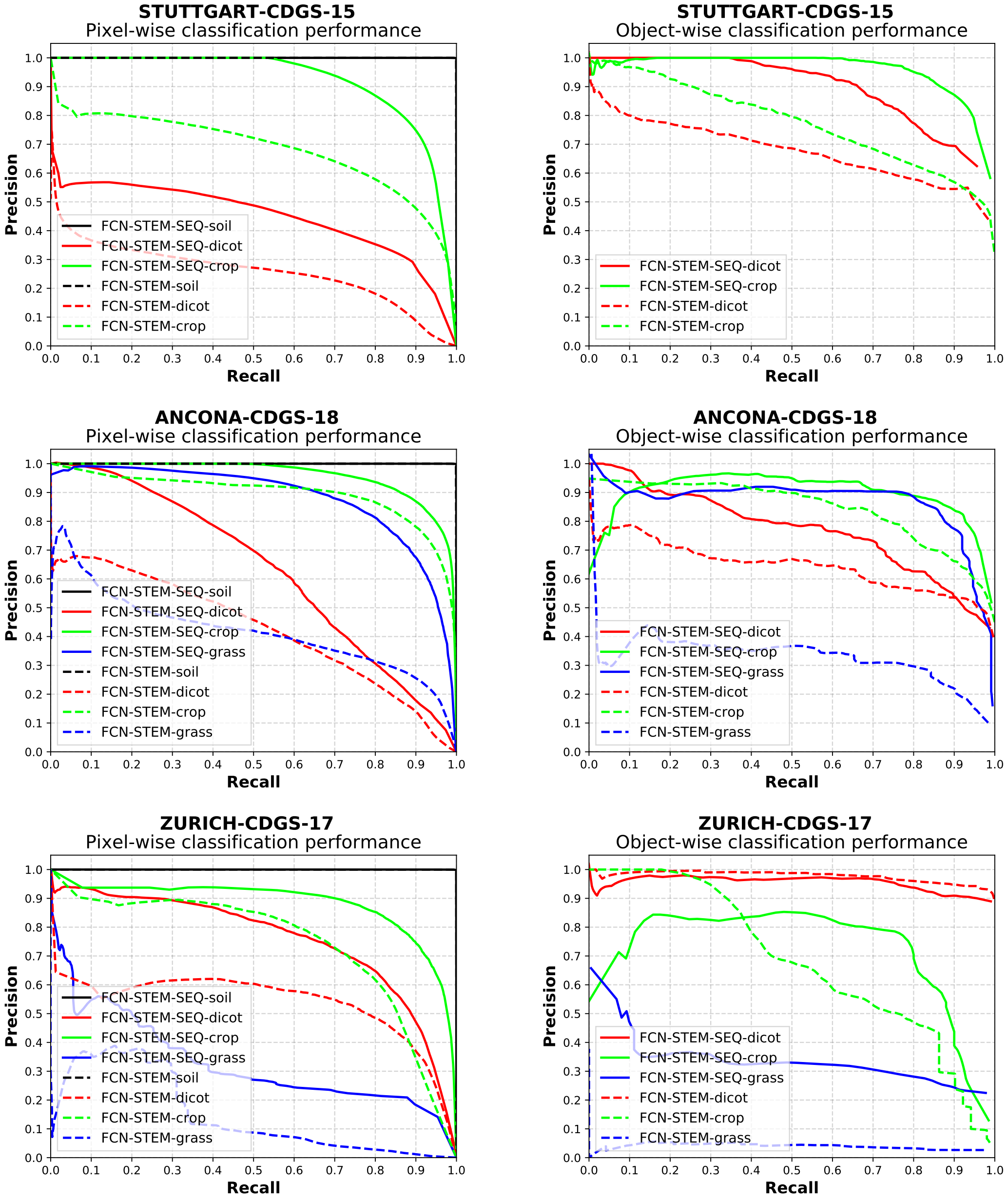


Figure 6.25: Precision-recall curves for the pixel-wise (left) and object-wise (right) crop-dicot-grass classification performance under changing field conditions. The curves illustrate the superior generalization capabilities of FCN-SEQ-STEM.

SEQ-STEM approach trained on BONN-CW-16, autonomous weed control would be possible. Here, we reach an object-wise recall for the crop of 95 % at a precision of 80 %. For the other datasets, however, the overall performance is still not suitable for autonomous weed control. Here, the classifier first needs to be adapted to the local field conditions.

The comparison of FCN-SEQ and FCN also suggests that the additional exploitation of spatio-temporal features leads to a better classification performance and better generalization to new and previously unseen field environments. Here, the relative gain for the sequential FCN-SEQ approach is about 14 % in terms of average F1-score. Thus, this experiment shows the superior generalization capabilities of our proposed FCN-SEQ-STEM and FCN-SEQ approaches and demonstrates the impact to the performance when exploiting a sequence of images with our proposed sequential module.

In Figure 6.26, we illustrate the qualitative results of our approach for all datasets, respectively. Overall, we observe that the plant classification for all classes is visually proper. However, there are small areas, particularly on crop plants and grass weeds, which are falsely classified as dicotyl weed. Since the total area of dicotyl weeds is small compared to the crops, even small error regions in the crop lead to a substantial drop in the pixel-wise classification metrics for dicotyl weeds. These results explain the low precision achieved for dicotyl weeds across all tested datasets. Note that the output used for the evaluation is the raw prediction by our approach, and no further postprocessing such as spatial smoothing is performed. By performing this postprocessing, we could improve the performance substantially due to the error source as mentioned above.

Analogous to the previous experiments under similar conditions, we evaluate the performance when using a shared encoder with two task-specific decoders. Therefore, we again compare the relative performance between FCN and FCN-STEM as well as between FCN-SEQ and FCN-SEQ-STEM. Also, under changing conditions, we observe a performance boost for the networks using two task-specific decoders. On average, the performance of the single-encoder-decoder networks FCN and FCN-SEQ drops around 6 % in average F1-score compared to the multi-tasks networks. These results support our conclusion from the previous experiment under similar conditions that the stem detection task loss also leads to the extraction of more descriptive features for the plant classification.

6.7.3 Conclusions for the Crop-Dicot-Grass Classification and Stem Detection Experiments

In total, we draw the following conclusions from the results in this section:

First, our approaches for joint plant classification and stem detection, i.e., FCN-STEM and FCN-SEQ-STEM, can provide suitable performance for plant-specific treatments wit high precision under similar field conditions. The stem detection works properly and provides the stem locations within a spatial precision of around 2 mm-4 mm.

Second, as for the crop-weed classification experiments, the results in this section support our claim of superior generalization capabilities for the fully convolutional

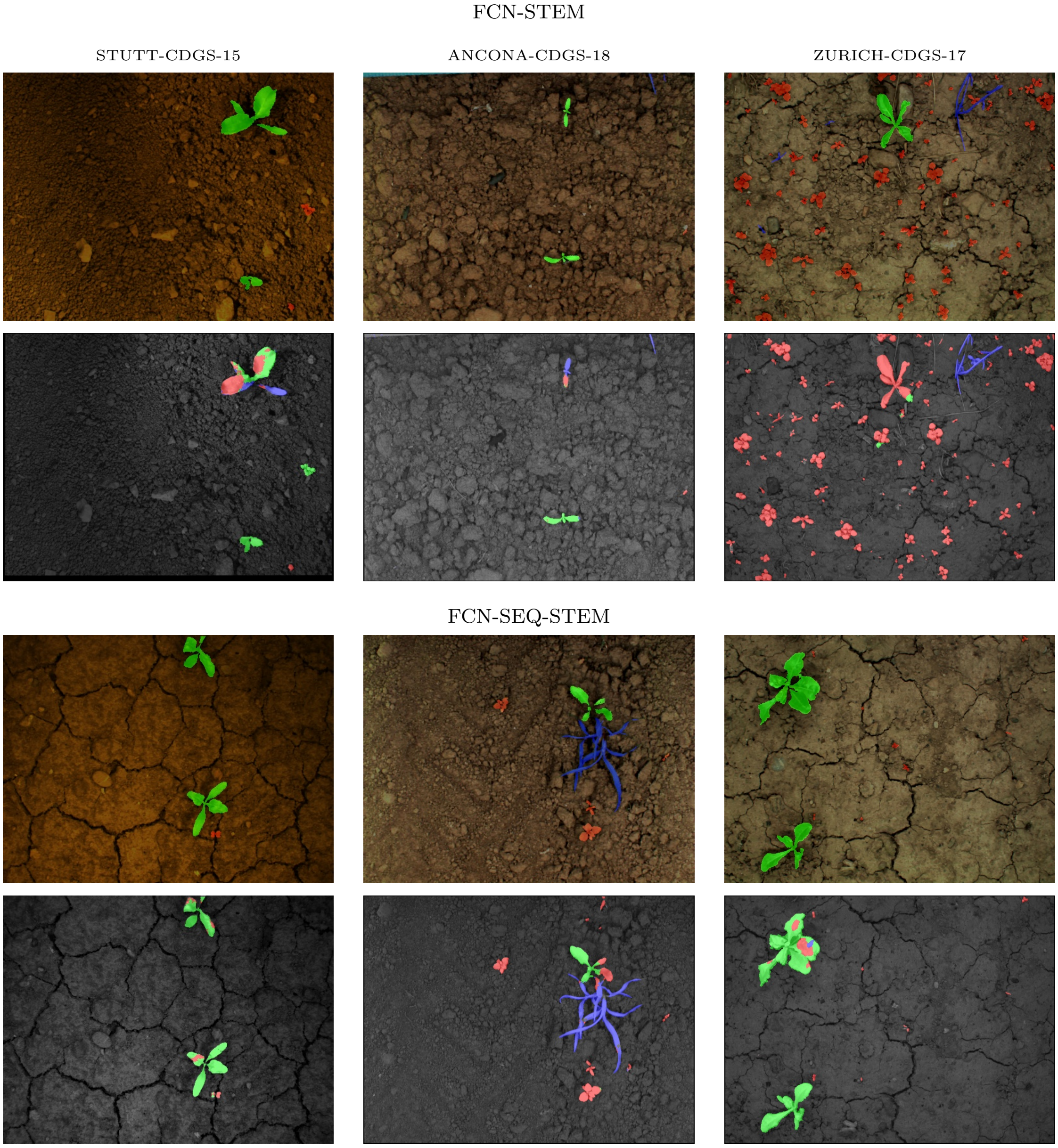


Figure 6.26: Qualitative results for the crop-dicot-grass classification performance of FCN-STEM and FCN-SEQ-STEM under changing field conditions. We show a representative example per approach and per dataset. Top rows: RGB image. Middle rows: ground truth overlayed on RGB image. Bottom rows: predictions overlayed on NIR image, where crop plants (green), dicotyl weeds (red), and grass weeds (blue) represent the pixel-wise classification.

neural network approaches exploiting the plant arrangement signal. In our experiments, however, the FCN-SEQ-STEM approach does not reach a performance level that is suitable for autonomous field intervention. The most probable reason for that is that the training dataset is too small to cover enough diversity to extract general enough features.

Third, the use of task-specific decoders that share a single encoder for the feature extraction aids the performance for both tasks, the plant classification, and the stem detection.

6.8 Supervised Classifier Transfer in the Context of Labeling Effort

In the previous experiments in Section 6.6 and Section 6.7, we have shown that the performance, but especially the reliability of the performance in new and changing field conditions suffers when the classifiers are deployed in new field environments. Even by exploiting the spatial arrangement of plants, it is not always possible to reliably obtain satisfactory results that would enable field robots to perform autonomous weed control.

In this experiment, we aim at evaluating the transferability of our classification models. We measure the required effort to adapt a model to a new dataset containing different environmental conditions. A typically applied case in practice is to use a particular training database training the classification models. If we now send the robot to a new field, it first collects new data, which is labeled to adapt the classifier to the local conditions. The question we tackle in this section is: How much labeling effort is needed to adapt our approaches to provide satisfactory results?

Therefore, we investigate our classifiers FCN, FCN-SEQ, and RF-GC concerning their adaptability to new data when only training on a small amount of labeled data from the new field environment. We use the BONN-CW-16 data to initially train models and investigate their crop-weed classification performance on the BONN-CW-17, STUTT-CW-15, ANCONA-CW-18, and ZURICH-CW-16 data sets after re-training the models with 1, 5, 10, 25, 50, and 100 additional labeled images.

Figure 6.27 illustrates the evolution of the average F1-score of the pixel-wise crop-weed classification performance after re-training the models on additional images from the targeted field domain. We report the performance on the remaining images of the respective test datasets. We see two noteworthy outcomes from this experiment.

As the first outcome of this experiment, the semi-supervised random forest-based RF-GC approach that combines a visual and geometric classifier only requires a little number of five additional images to substantially improve its performance on the target data sets substantially. Except for the field in Zurich, RF-GC already achieves a comparable or even better performance to the fully convolutional neural network approaches. This is remarkable since, in all previous experiments, the random forest performed worse than the fully convolutional neural network approaches. On average, RF-GC achieves a performance gain of about 30 % average F1-score if we leave the

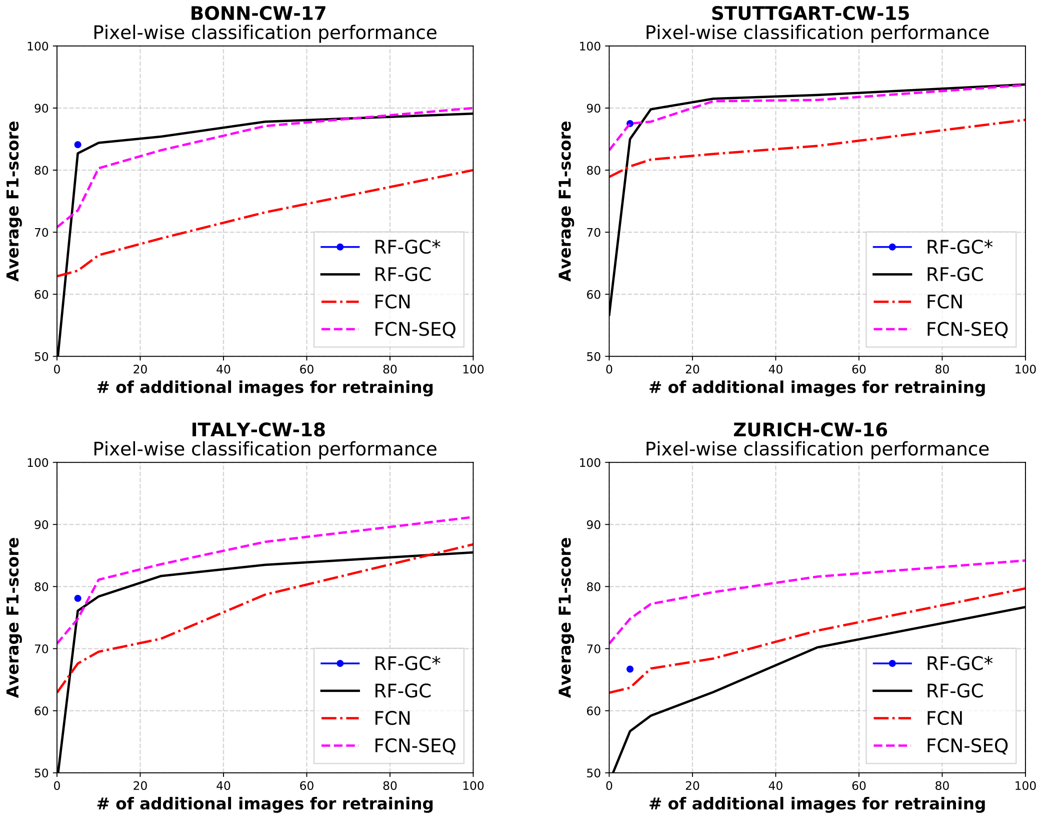


Figure 6.27: Precision-recall curves for the pixel-wise crop-weed classification performance after re-training the models on additional images from the targeted field domain.

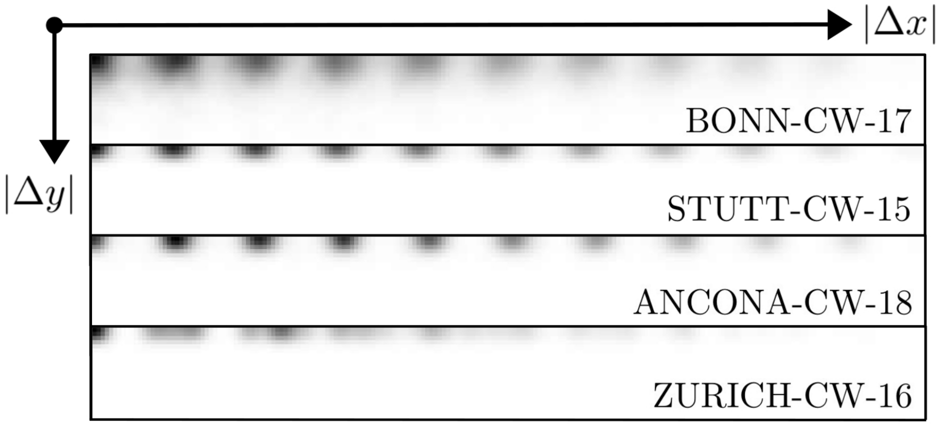


Figure 6.28: Plant arrangement model $p(\boldsymbol{d} \mid \omega_\mathsf{c})$ according to Equation (4.19). In this illustration, we learned the respective $p(\boldsymbol{d} \mid \omega_\mathsf{c})$ for each dataset only considering ground truth data, i.e., the manually labeled plants in the images. The black color refers to the probability mass for the distribution of the coordinate difference along and perpendicular to the crop row.

ZURICH-CW-16 dataset aside. Besides, the random forest can slightly increase its performance by including further data, but not as much as after the first five images.

There are two reasons for this substantial performance growth. First, the RF-CAS approach starts from a rather low performance, which is caused by initialization problems of the geometric classifier on the test data, in the case without re-training. Therefore, an increase due to new training data is likely. The fact, however, that the random forest with five additional images already provides better results than the neural network approaches is because of a successful initialization of the geometric classifier. Exploiting the plant arrangement compensates for the lack of generalization capability of the visual classifier. Second, over time, the visual classifier is adapted and provides better predictions that further stabilize the whole system.

We investigate why the performance on the ZURICH-CW-16 data does follow this performance pattern. We analyze the quality of the relative plant arrangement and conclude that the reason for the lower performance on the ZURICH-CW-16 dataset is the higher variance for the spacing of plants along the crop row. Figure 6.28 shows the arrangement models $p(\boldsymbol{d} \mid \omega_\mathsf{c})$ learned by the manually labeled ground truth data. From the distribution of the probability mass, we can conclude the precision of the geometric distribution of the plants. An equidistant distribution of the plants leads to a multimodal distribution of coordinate differences along the row axis where the individual modes have an equal distance between each other, see Figure 6.28 (ANCONA-CW-18). In contrast, an irregular distribution of the plants results in a distribution of a with considerably more dispersion, see Figure 6.28 (ZURICH-CW-16). Here, the lower quality of the intra-row spacing in Zurich leads to smaller support by the geometric classifier. Nevertheless, the RF-CAS approach provides effective transfer capabilities with small amounts of training data, in case of a particular quality of the plant's row spacing.

Regarding the second outcome of this experiment, we observe a constant perfor-

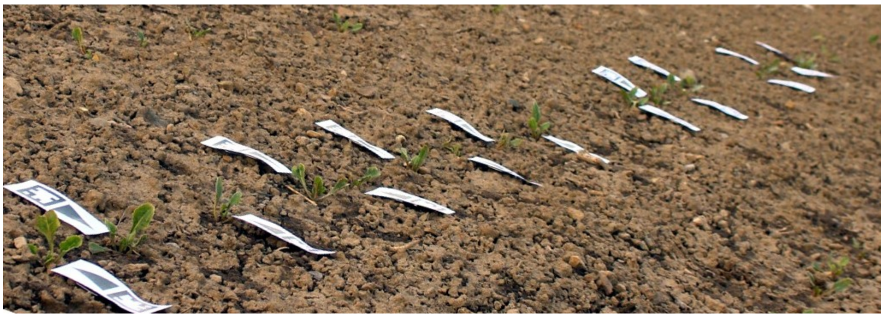


Figure 6.29: Labeling of the crop plants (sugar beets) with markers placed next to the plant.

mance improvement when the classifier has access to more and more training data coming from the targeted field domain for the fully convolutional neural network approaches. In particular, the sequential FCN-SEQ approach always performs better than the non-sequential FCN approach. This is to be expected, as it already starts from a higher performance just by applying the pre-trained model on the test data. However, if we look at the slope of the performance concerning the additional training data available, the curve of the FCN-SEQ approach reveals a tendency that the sequential approach can better exploit small amounts of data. Compared to the FCN curve, the performance increase is slightly better for the re-training with up to 25 images.

We conclude that the approaches RF-GC and FCN-SEQ, which additionally exploit the spatial arrangement of the plants, serve a more effective adaption to new and changing field conditions under the view of the needed labeling effort. FCN-SEQ reaches at least 85 % average F1-score on the test data by using 100 images for re-training. In three of four test cases, it reaches 90 % average F1-score.

6.8.1 Comparison Under Minimal Labeling Effort

The noteworthy results for RF-CAS raise the question of whether this approach even requires a pre-trained visual classifier to perform well on the test data. We perform the same experiment again with the RF-GC approach and in this case only use a few images, which were originally intended for the re-training, but in this experiment, to initialize the visual and geometric classifier entirely from scratch.

In this experiment, we reduce the labeling effort to its extreme. We target a labeling effort of approximately one minute for a human and do not consider any pre-trained classifier. We achieve this one-minute labeling effort by placing printed markers next to a set of sugar beet plants at the beginning of the row. Figure 6.29 illustrates how the one-minute in-field labeling works. We can place around 10-15 markers within a minute, which corresponds to approx. 2 m-3 m of sugar beets along a row. We find the markers in the images and assign the label “crop” to detected vegetation based on a distance threshold, all other vegetation is considered to belong to the “weed” class. Based on this information, we can initialize the plant arrangement model and start training the visual classifier of RF-GC and can train the random forest model of RF-GC. We perform the procedure based on markers for the BONN-CW-17 and

ANCONA-CW-18 dataset. For the STUTT-CW-15 and ZURICH-CW-16 dataset, we use ten labeled images representing the first 3 m of the crop row to approximate the marker-based labeling.

The term "RF-GC ⋆" in Figure 6.27 refers to the performance obtained under the aforementioned setup. For all test datasets, the achieved performance of RF-GC ⋆ is comparable or even higher than the performance of RF-GC using a re-trained visual classifier. This experiment shows the potential of the RF-CAS approach, which exploits its geometric classifier to adapt the visual random forest-based classifier on the fly. For the field in Zurich, it seems that the re-trained classier even harms the performance of the systems. We argue that the wrong predictions of the visual classifier of the RF-CAS approach are not in line with one provided by the geometric classifier and, thus, lead to more false predictions in total.

We conclude that the RF-CAS approach serves excellent capabilities to rapidly adapt to new field environments, given that the quality of the spacing of the crop plants is "good enough".

6.8.2 Conclusions for Supervised Classifier Transfer in the Context of Labeling Effort

We draw the following conclusions from the results in this section:

First, under the view of the needed labeling effort, our proposed RF-GC approach provides the best performance and adapts the best to the current situation in a new field environment. However, it relies on a sufficient quality of the plant spacing along the crop row.

Second, the sequential FCN-SEQ approach can better exploit smaller amounts of data to adapt to new and changing field conditions compared to its non-sequential variant FCN.

Third, both fully convolutional neural network approaches can be re-trained to provide a high-quality classification in new field conditions by using around 100 images from the targeted field domain.

6.9 UAV-Based Plant Classification for Automated Crop Monitoring

We design the next evaluation in this section to illustrate the performance of our UAV-based plant classification systems. We evaluate performance in two ways. First, we analyze the classification performance for the automatic computation of the spatial distribution of plants and weeds. This information is important for the scheduling of weed control operations and for the provision of application maps, which are a prerequisite for the application of the minimum amount of chemicals required for the current situation in the field. Second, we evaluate the counting performance with which we can derive the number of actual plants in the field. The number of plants in the field is an important trait for seed quality and can also be used to estimate the expected yield. For both applications, we show that our proposed approaches are capable of providing accurate maps of the crop plants and weeds on a per-plant basis, also including weeds located in the intra-row space.

For all UAV experiments, we solely consider RGB images as the input to the classification systems. Using UAV data instead of data recorded with a UGV, however, is more challenging, as the imagery is naturally exposed to varying lighting conditions. Another difference between the UAV and UGV data is the camera footprint on the field. UAV images are much larger with 12-21 mega-pixels and cover a larger area of the field with only one image. The image data for the application used in this section does not have to be processed online during data acquisition. Thus, for image processing, we do not focus on runtime and, therefore, perform no downscaling of the images as in the case of UGVs.

We separate the experiment in this section into five subsections. In Section 6.9.1 and Section 6.9.1, we evaluate the pixel-wise crop-weed classification for high- and low-resolution imagery, respectively. In Section 6.9.3, we evaluate the performance in the view of multi-species classification for estimating the spatial distribution of different weed types. In Section 6.9.3, we evaluate the effectiveness of the UAV-specific, geometric features used in RF-UAV that encode information about the plant arrangement. For these four experiments, we evaluate the random forest-based RF-UAV approach and our FCN-UAV approach in terms of the plant classification. In Section 6.10, we evaluate the performance to estimate the number of actual crop plants in the field. For the plant counting experiments, we use the FCN-UAV-STEM approach.

Since the actual ground resolution of UAV images is generally coarser compared to UGV images, we avoid downsampling the image data to a suitable resolution for the network as for the UGVs. Consequently, we split the images into smaller chunks of size 512 × 512 pixels. Furthermore, the training procedure differs from that during the deployment phase of the classifier. Throughout the training phase, we randomly extract image patches of size 512 × 512 pixels and randomly rotate them to achieve learned features that are robust against different orientations of the crop rows in the data. In the classifier's deployment phase, we divide an input image into 512×512 pixel areas and select the overlap such that we can drag the patches without resizing the original image. After the prediction of the image patches, we stitch them together again to the original input image size. For the overlapping regions, we average the single probabilities for one pixel across all classes and then assign the average probability.

6.9.1 Classification Performance for High-Resolution Imagery

We design these experiments to demonstrate that our approaches RF-UAV and FCN-UAV can classify plants and weeds in high-resolution UAV images. With a high resolution, we mean a ground sampling distance of about 1 mm. This resolution is comparable to the one used in the UGV experiments in this chapter.

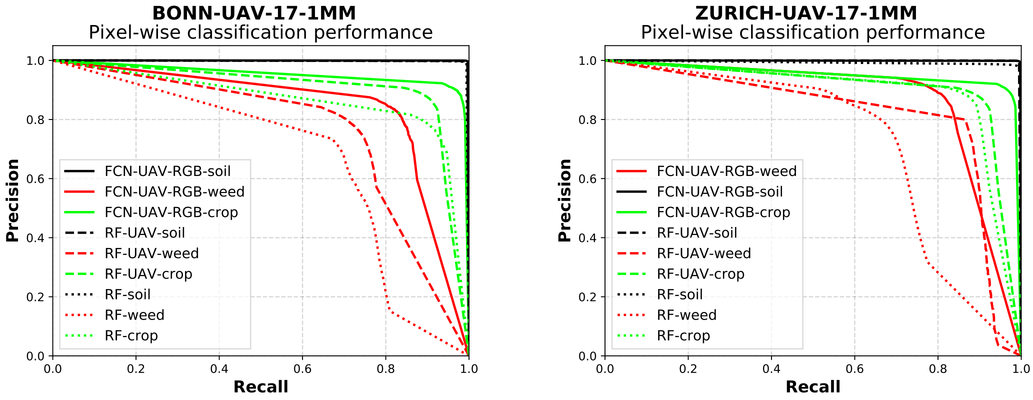


Figure 6.30: Precision-recall curves for the pixel-wise UAV-based crop-weed classification performance under similar field conditions. Left: classifier trained on BONN-UAV-17-1MM and deployed on BONN-UAV-17-1MM test split. Right: classifier trained on ZURICH-UAV-17-1MM and deployed on ZURICH-UAV-17-1MM test split. Green refers to crop plants and red refers to weeds.

Table 6.21: Pixel-wise crop-weed classification performance obtained through our RF-UAV and FCN-UAV approach under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-UAV-17-1MM (25 %) Deployment: BONN-UAV-17-1MM (70 %)										
FCN-UAV	92.0	87.9	88.8	93.6	92.8	94.4	83.0	82.9	83.2	99.3
RF-UAV	87.5	84.4	79.6	88.0	87.8	88.2	75.6	81.0	70.9	98.9
Training: ZURICH-UAV-17-1MM (25 %) Deployment: ZURICH-UAV-17-1MM (70 %)										
FCN-UAV	93.0	91.0	88.8	94.8	93.5	96.2	84.7	88.5	81.3	99.3
RF-UAV	89.9	84.9	86.7	87.9	87.7	88.2	83.5	82.0	85.1	98.2

The datasets used for this evaluation are BONN-UAV-17-1MM and ZURICH-UAV-17-1MM. Both datasets consist of around 90 pixel-wise annotated images containing sugar beets and a substantial amount of weeds observed in different growth stages and under different weather conditions. For the high-resolution UAV imagery, we perform two experiments. First, we evaluate the performance under similar field conditions by analyzing the intra-dataset performance within the BONN-UAV-17-1MM and ZURICH-UAV-17-1MM datasets, respectively. Second, we evaluate the performance under changing field conditions by analyzing the cross-dataset performance between BONN-UAV-17-1MM and ZURICH-UAV-17-1MM.

Evaluation under similar field conditions For the evaluation under similar field conditions, we train the classifiers on a random 25 % split of the images and test them on

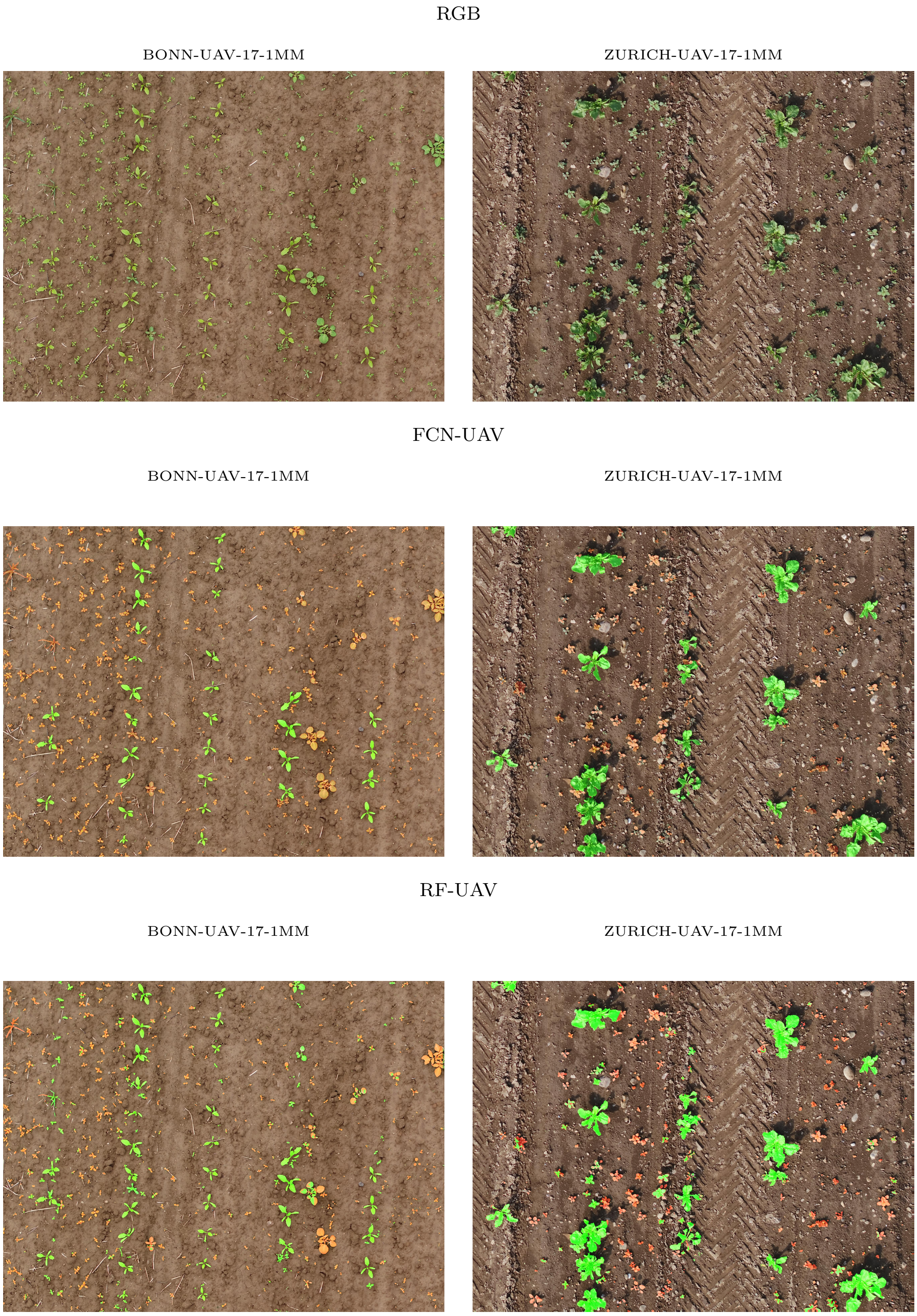


Figure 6.31: Qualitative results for the pixel-wise UAV-based crop-weed classification performance of FCN-UAV and RF-UAV under similar field conditions. We show a representative example per approach and per dataset. The FCN-UAV approach provides a high-quality crop-weed classification, whereas the RF-UAV approach produces more wrong classifications, especially in the crop row area.

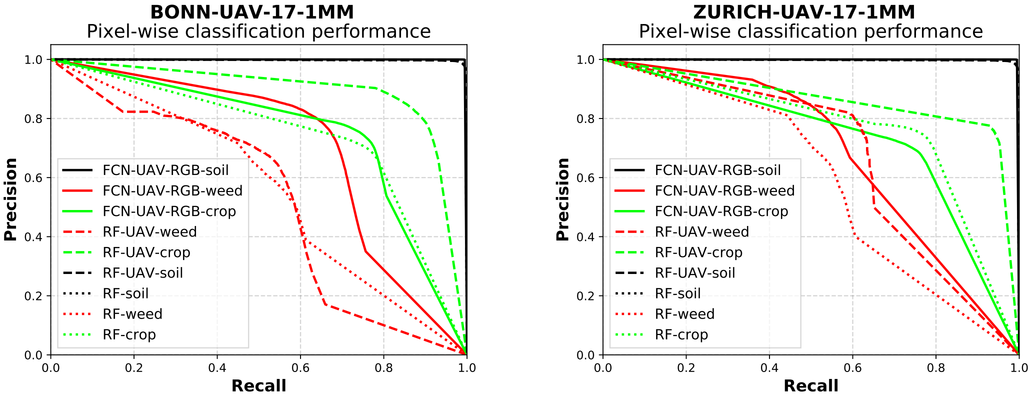


Figure 6.32: Precision-recall curves for the pixel-wise crop-weed classification performance under changing field conditions. Left: classifier trained on BONN-UAV-17-1MM and deployed on ZURICH-UAV-17-1MM. Right: classifier trained on ZURICH-UAV-17-1MM and deployed on BONN-UAV-17-1MM.

70 % of the BONN-UAV-17-1MM and ZURICH-UAV-17-1MM datasets, respectively. We use the remaining 5 % as validation portion data to perform early stopping. Figure 6.30 depicts the obtained precision-recall curves of the respective test splits for the pixel-wise crop-weed classification performance. The comparison of the two approaches shows a significant advantage of FCN-UAV for both crops and weeds. Overall, FCN-UAV performs around 4 %-5 % better than RF-UAV with respect to the average F1-score across all classes, see also Table 6.21. One reason for this is an almost constant high precision of around 95 % for crop plants with the recall interval of 90 %-97 %. Also, for weeds, FCN-UAV performs around 8 % better in terms of the class-specific F1-score on the BONN-UAV-17-1MM dataset. These results convey that our fully convolutional neural network approach is better suited for crop-weed classification under similar field conditions.

The qualitative results depicted in Figure 6.31 show the reasons for the different performance of the evaluated approaches. FCN-UAV properly separates overlapping weeds and crop plants for both datasets, whereas RF-UAV has notably more missclassifications for weeds located near the crop rows. We argue that the line model feature encoding the distance to the crop row pushes keypoints and objects located close to row to be classified as a crop. Thus, we investigated the classification without using the line model feature. However, the precision-recall curve for the basic RF-CAS approach in Figure 6.30 shows that not using geometric features leads to even more missclassifications for weeds (mostly located between the rows).

Evaluation under changing field conditions For the evaluation under changing field conditions, we use 95 % of the BONN-UAV-17-1MM dataset for the training and report the achieved performance for the entire ZURICH-UAV-17-1MM dataset. For completeness, we also perform the same experiment in the other direction. We use the remaining 5 % as validation portion data to perform early stopping. Figure 6.32

Table 6.22: Pixel-wise crop-weed classification performance obtained through our RF-UAV and FCN-UAV approach under changing field conditions. We report the class-wise and average F1-score (F1), precision (P) and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	**Average**			**Crop**			**Weed**			**Soil**
	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-UAV-17-1MM (25 %) Deployment: BONN-UAV-17-1MM (70 %)										
FCN-UAV	81.6	68.5	78.0	74.5	73.9	75.1	70.9	63.1	80.8	99.3
RF-UAV	80.7	72.8	71.9	84.8	80.1	90.1	59.0	65.4	53.7	98.2
Training: ZURICH-UAV-17-1MM (25 %) Deployment: ZURICH-UAV-17-1MM (70 %)										
FCN-UAV	79.9	71.9	68.6	73.0	72.4	73.6	67.2	71.3	63.5	99.5
RF-UAV	81.1	69.3	76.1	84.0	78.6	90.2	60.9	59.9	62.0	98.2

depicts the resulting precision-recall curves for the pixel-wise UAV-based crop-weed classification performance of RF-UAV and FCN-UAV under changing field conditions.

Under these conditions, the RF-UAV approach is not able to provide suitable results, as the threshold based vegetation classification, which is based on the threshold learned from the training data, fails. To compare the performance concerning the crop-weed classification, we adjust the threshold on one image of each test data set, respectively, and report the performance under these conditions.

Figure 6.32 depicts the resulting precision-recall curves for the pixel-wise crop-weed classification performance of RF-UAV and FCN-UAV under changing field conditions. Table 6.22 summarizes the achieved performance for labeling with the most likely class. First, we see that the overall performance of both approaches is lower than the one under similar field conditions. This observation is in line with the evaluation for UGVs, see Section 6.5.2. Only the RF-CAS approach can provide useful results for the classification of crop plants. The explicit modeling of geometric features is mainly responsible for the recall of 90 % at a precision close to 80 % for the crop class on both datasets. The FCN-UAV approach cannot provide this level of performance. Since the architecture is not explicitly designed for the extraction of the field geometry, the color information might be considered to be too important. This, in turn, leads to a loss in performance when the visual appearance of the plants and soil changes. To test this hypothesis, we filter the FCN-UAV results within an additional postprocessing step. Based on the classification result, we first extract the crop row information using the method described in Section 4.5.1. Subsequently, we re-label all falsely predicted crop pixels that have a distance of >10 cm to the extracted crop row as weed pixels. This procedure increases the precision for the crop by about 10 % and the recall for weed by about 20 %. Thus, the geometric information has the potential to improve the purely visual classification.

RF-UAV achieves low F1-scores of around 60 % for weed on both datasets. The primary reason for that is the F1-score for the soil of 98.2 %. As the majority of pixels

Table 6.23: Pixel-wise crop-weed classification performance obtained through our RF-UAV and FCN-UAV approach for low-resolution imagery under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-UAV-17-5MM (45 %) Deployment: BONN-UAV-17-5MM (50 %)										
FCN-UAV	93.4	91.1	90.1	93.9	91.9	96.0	87.0	90.2	84.1	99.3
RF-UAV	90.8	88.5	85.9	92.5	89.8	95.3	81.5	87.1	76.5	98.6

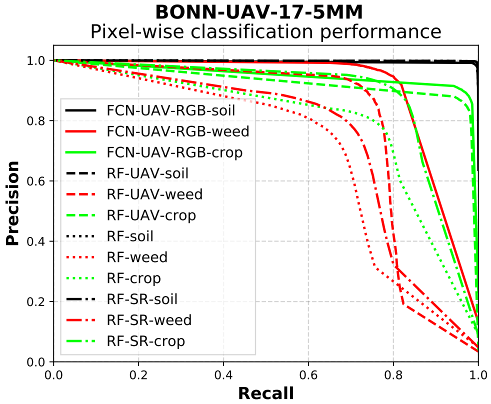


Figure 6.33: Precision-recall curves for the pixel-wise UAV-based crop-weed classification performance at 5 mm ground sampling distance. The FCN-UAV approaches provides the best performance. The RF-UAV approach profits from the use of the geometric features and provides the second-best performance.

belong to soil, a small percentage-wise decrease in the performance for soil can have a substantial effect on the vegetation classes. Under this view, the FCN-UAV approach provides better performance compared to the threshold-based vegetation classification of RF-UAV. The threshold-based vegetation classification suffers from the sunny conditions in the ZURICH-UAV-17-1MM data.

We draw the conclusions that explicitly modeling the geometric features helps to bridge the performance loss when deploying the classifiers in new and unseen field environments. Furthermore, fully convolutional neural networks provide a more robust separation of soil and vegetation under changing field conditions.

6.9.2 Classification Performance for Low-Resolution Imagery

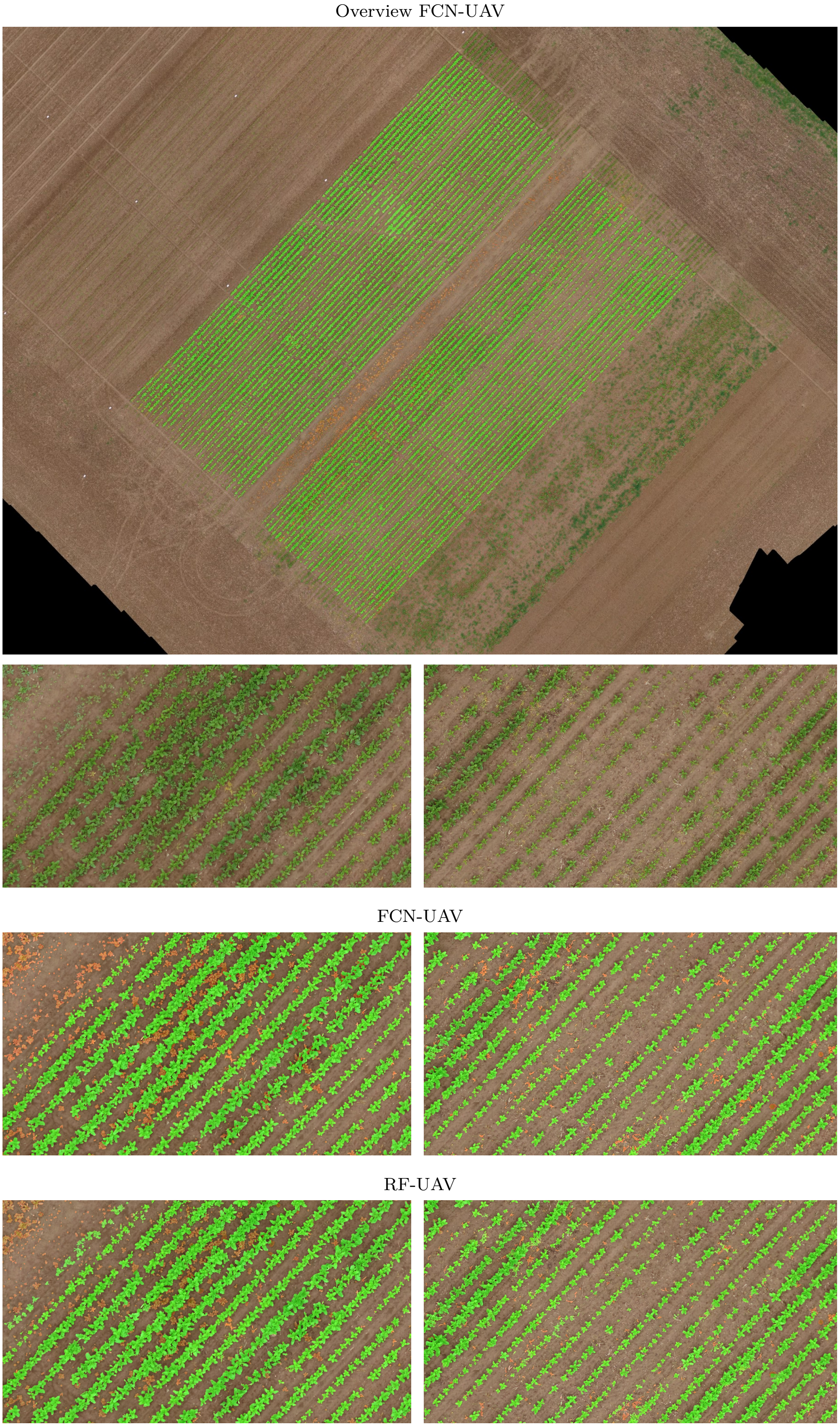
In this section, we evaluate the performance of FCN-UAV and RF-UAV for a lower ground resolution of the image data. While we perform the experiments in the previous section with a ground resolution of around 1 mm per pixel, we now analyze the performance of the crop-weed classification with a ground resolution of around 5 mm per pixel. A lower resolution reflects more challenging conditions for the classifiers, as they have to differentiate between crop plants and weeds based on less pixel information. From the application's point of view, however, a lower resolution is desirable, as it leads to faster coverage during flight. Note that the observed area increases to the square of the ground sampling distance when using the same acquisition setup.

We perform this experiment on the BONN-UAV-17-5MM dataset, which we describe in Section 3.2.3. We train the RF-UAV, and FCN-UAV approaches on nine manually selected image regions containing sugar beets and a substantial amount of weeds and test it on another ten test patches for which we have pixel-wise ground truth information. We use one patch as validation data to perform early stopping. Figure 6.33 depicts the resulting precision-recall curves for the pixel-wise UAV-based crop-weed classification performance of RF-UAV and FCN-UAV at 5 mm ground sampling distance.

Figure 6.34 shows an overview of the classified field obtained by the FCN-UAV approach. We can observe that the fully convolutional neural network approach qualitatively provides an appropriate accuracy for crop-weed classification. Figure 6.33 depicts the obtained precision-recall curves for the pixel-wise crop-weed classification concerning the ten test patches.

Table 6.23 summarizes that both models classify the majority of crops correctly and obtain a high recall of >95 % at a precision of around >90 %. The shapes of the precision-recall curves indicate a stable prediction of the crop plants over the entire recall range of 0-95 %. The performance for the weed class, however, differs between the random forest and the fully convolutional neural network. Here, the precision-recall curve for the neural network starts decreasing later, i.e., for higher recall values. Thus, for weeds, we see an advantage for the FCN-UAV approach. Concerning the performance for labeling with the most likely class, the fully convolutional neural network approach achieves a gain of around 7 % in terms of thew F1-score.

For a qualitative inspection, Figure 6.34 depicts zoomed views of analyzed field regions concerning the spatial distribution of crop plants and weeds. The FCN-UAV approach can better classify the weeds that are located close to crop plants compared to RF-UAV. Thus, it classifies more of the actual weed canopy correctly. The RF-UAV approach cannot correctly classify all weeds that grow close to or overlap with crop plants. It tends more to predict the class crop if objects or keypoints are located close to crop row. To reason about this observation, we investigated the feature importance provided by the training procedures of the random forest. Here, we find that the line model feature (see Section 4.5.1) is the third most important feature for the object-based features and the second most important feature for the keypoint-based features. Therefore, we argue that the crop row feature has a too strong influence on the prediction of these cases. Figure 6.34 (bottom row, left) illustrates that the RF-UAV approach wrongly classifies a substantial number of weeds located next to the left-most crop row. The reason is that the crop row extraction falsely detects another potential crop row. In consequence, the keypoint and objects along this falsely detect



Overview FCN-UAV

FCN-UAV

RF-UAV

Figure 6.34: Overview of the crop-weed classification result on the BONN-UAV-17-5MM dataset obtained by FCN-UAV and zoomed view for classification results obtained by FCN-UAV and RF-UAV

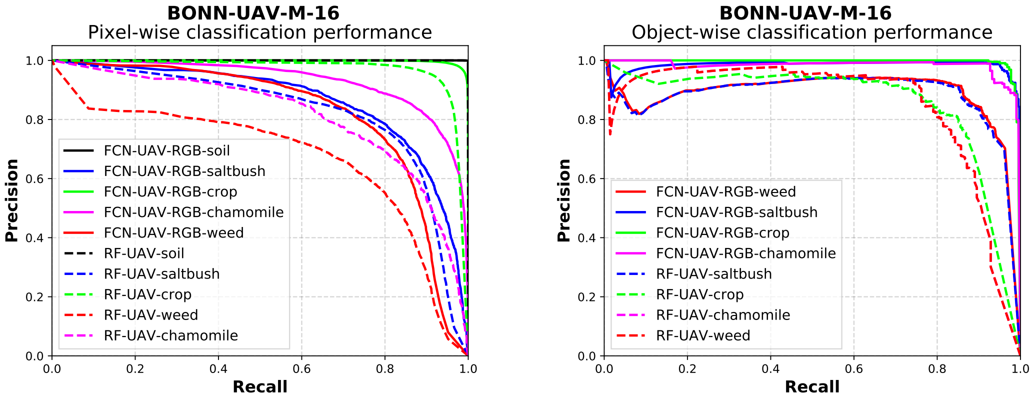


Figure 6.35: Precision-recall curves for the pixel-wise UAV-based multi-species classification on the BONN-UAV-M-16 dataset.

crop row get a wrong value for the line model features. We further investigated the performance of the RF-UAV approach when reducing the size of the patches to up to 10×10 pixels. However, the steering to a smaller patch size did not lead to an increase in the performance.

Thus, we conclude that the performance of the RF-UAV approach can be limited through a lower ground resolution and that the FCN-UAV is well suited for the estimation of crop-weed maps for both high- and low-resolution imagery.

6.9.3 Multi-Species Classification

In this section, we target the detection of common weed species in sugar beet fields in Northern Europe, which is an important problem and a challenging task for precision farming systems. The information about the spatial distribution of different weed species in a field can be exploited to trigger weed control tasks and to plan the most appropriate mixture of herbicides, given the current species distribution in the field. We design this experiment to demonstrate that our approaches RF-UAV and FCN are capable of classifying sugar beets and different weed species, which are common on sugar beet farms.

Therefore, we analyze the multi-species classification performance on the BONN-UAV-M-16 dataset, which we describe in Section 3.2.3. The dataset consists of 20 fully annotated images containing sugar beets and several weeds species that we labeled manually as sugar beets, saltbush, chamomile, other weeds, and soil. We train the classifiers on nine images, test them on ten images, and use one image as validation data to perform early stopping. Note that for the RF-UAV approach, we do not exploit the line model features as no crop row structure is present in this dataset.

Figure 6.35 depicts the precision-recall curves for the multi-species classification at the pixel level as well as at the object level. Considering the pixel-wise performance, it shows that FCN-UAV and RF-UAV can achieve class-wise recalls greater than 90 % depending on the selected threshold for the label assignment. The class labeling based

Table 6.24: Pixel-wise multi-class classification performance btained through RF-UAV and FCN-UAV approach under similar field conditions. We report the class-wise and average F1-score (F1), precision (P), and recall (R) for a labeling with the most likely class in percent according to Equation (2.23). The term training refers to the used training data for a particular experiment, whereas the term deployment refers to the test dataset.

Approach	Average			Crop			Chamomile			Saltbush			Weed			Soil
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Training: BONN-UAV-M-16 (45 %) Deployment: BONN-UAV-M-16 (50 %)																
FCN-UAV	87.6	89.1	94.6	97.3	97.2	97.5	85.9	80.9	91.6	79.4	76.1	83.0	77.5	75.6	79.5	99.3
RF-UAV	82.0	81.1	86.8	92.9	92.8	93.1	74.5	69.4	80.4	78.6	75.7	81.8	67.3	60.2	76.4	98.2

on the predicted maximum confidences of the FCN-UAV and RF-UAV approach leads to the results that are listed in Table 6.24. In terms of the average F1-score across all classes, the fully convolutional neural network approach achieves a 6 % better performance at the pixel level. This performance gain is mainly caused by the better results for chamomile and other weeds. Generally, the overall precision of both classification systems suffers from the obtained recall for other weeds ranging from 76 % for RF-UAV to 80 % for FCN-UAV. This result is affected by having a small number of examples within the datasets and probably by a higher intra-class variance, since all actual other weeds, which occur in this dataset, are represented by this class. More datasets with different weed types are needed to clarify that. If we only focus on crop-weed classification, the overall F1-Score increases by 9 % for both approaches to 97 % for FCN-UAV and 91 % for RF-UAV.

Object-wise, the FCN-UAV approach achieves a substantially higher for the sugar beet class with around 17 %. Moreover, it can predict the different weed species with a higher precision, leading to better overall performance. We conclude that the fully convolutional neural network approach can extract more descriptive features to distinguish different weed species and, thus, can better deal with the small number of training images.

Figure 6.36 depicts analyzed images for the multi-class classification obtained by the RF-UAV and FCN-UAV approach. In this illustration, we do not show the ground truth image because a direct comparison is difficult to achieve visually. The visual inspection of the predictions reveals that most of the plants and weeds are classified correctly by the fully convolutional neural network approach, whereas the random forest produces more misclassifications in between the different weed species. This observation also coincides with the performance on object-level that is depicted in Figure 6.35 (right).

6.9.3.1 Impact of Geometric Features for RF-UAV

The additional geometric features, namely the line model feature for crop rows described in Section 4.5.1 and the spatial relationship features described in Section 4.5.2, represent the adoption of the RF-UAV to UAV images based on the basic RF-CAS approach. We design this experiment to demonstrate the impact in performance when using ge-

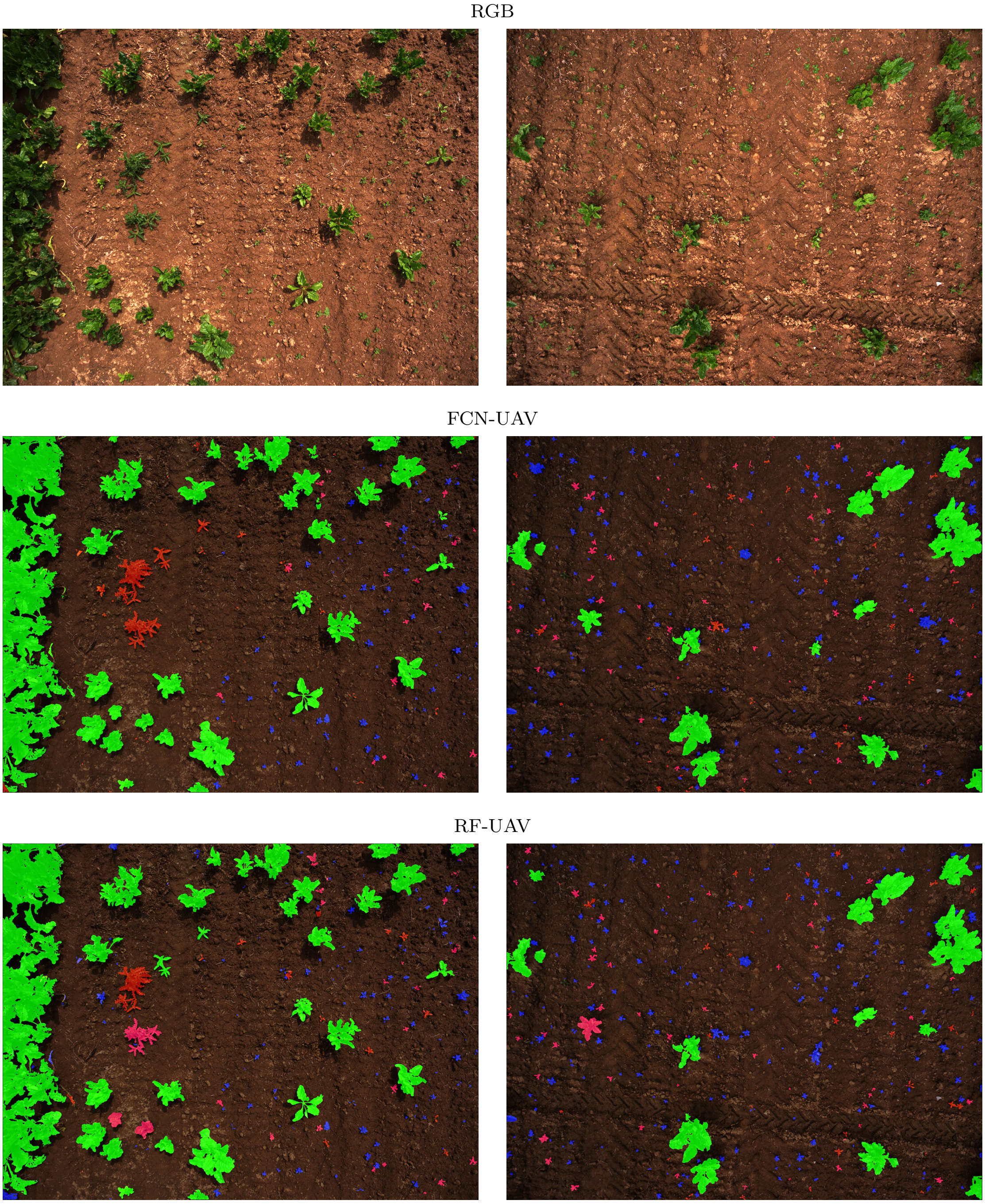


Figure 6.36: Multi-species classification results on the BONN-UAV-M-16 dataset. Example UAV images analyzed by our approaches RF-UAV and FCN-UAV. Different colors refer to the plant classification considering the classes crop in green, saltbush in blue, chamomile in magenta, other weed in red, and soil.

ometric features in our RF-UAV approach. Therefore, we compare the performance of the RF-CAS and RF-UAV approach on the high-resolution BONN-UAV-17-1MM and ZURICH-UAV-17-1MM datasets as well as on the low-resolution BONN-UAV-17-5MM dataset.

First, we analyze the impact of exploiting the geometric features for high-resolution images under similar field conditions. Figure 6.30 depicts the performance of RF-CAS and RF-UAV in terms of precision-recall curves for the BONN-UAV-17-1MM and ZURICH-UAV-17-1MM datasets. RF-UAV achieves substantially better performance for crop plants and weeds on both datasets. The geometric features lead to a performance gain of around 6 % average F1-score taking both datasets into account. The biggest gain is achieved for weed on the ZURICH-UAV-17-1MM dataset with around 10 % in F1-score. The RF-CAS approach suffers from the comparably large diversity in the respective UAV datasets. The exploitation of geometric features in RF-UAV, however, compensates for the performance loss and thereby improves the performance close to the level of the fully convolutional network.

Figure 6.32 depicts the precision-recall curves for the performance under changing field conditions. Compared to the purely visual random forest, RF-UAV obtains a better F1-score of around 12 % on the BONN-UAV-17-1MM and 7 % on the ZURICH-UAV-17-1MM dataset for the crop. Also, for weed, the performance increases up to 5 % F1-score when using geometric features. Concerning all tested approaches to these data, RF-UAV is the only method providing decent performance for the crop class under changing field conditions.

Our evaluation shows that for high-resolution imagery, the use of geometric features supports the classification based on visual features, as it improves the overall accuracy and precision, especially for the crop, on the tested datasets.

We observe the biggest gain in performance for the BONN-UAV-17-5MM dataset containing low-resolution images with a ground sampling distance of around 5 mm. Here, the detection of RF-CAS based only on visual appearance, i.e., features ignoring geometry, suffers from the comparably low ground resolution. Thus, geometric features become great supporters for the detection, as they are rather invariant to the image resolution. First, we add our proposed spatial relationship features to RF-CAS. The term RF-SR refers to the performance that is achieved in this case. We observe a performance gain of around 4 % for the weed and 6 % for the crop class. Adding the line model features on top further increases the performance. In total, the RF-UAV approach achieves a better F1-score of around 10 % for weed and 16 % for crop compared to RF-CAS. The corresponding precision-recall curves indicate that this amount is mainly caused by better detection of the crop class. Thus, the geometric features are key supporters for the crop-weed classification on low-resolution imagery.

We conclude that using geometric features for the classification task is an appropriate way to exploit spatial characteristics of plantation in agricultural field environments.

6.10 UAV-Based Plant Counting Under Harsh Conditions

We design the experiment in this section to evaluate the performance of the FCN-UAV-STEM approach in applying plant counting under harsh conditions. These experiments were conducted in collaboration with the Institut für Zuckerrübenforschung and ARGE NORD, who supported the experiment in data collection and field management. We demonstrate the effectiveness of FCN-UAV-STEM by performing the analysis of plant counting under hard conditions, as described in Section 6.10. The conditions include overlapping plants, high weed pressure, straw- and weed-overlaid and concealed plants in various growth stages, and images recorded under different weather conditions.

For this experiment, we use the GOETT-UAV-19 dataset described in Section 3.2.4. This dataset consists of three measurements of a field with 40 microplots each, i.e., 120 microplots in total, see Figure 3.17 for an illustration. Already at the early growth stage, the sugar beets overlap each other along the crop row. Individual plants appear as separate components in the image-space because they are covered and separated by straw or larger grown weeds.

We train our classification model FCN-UAV-STEM on five manually labeled microplots per measurement day, i.e., 15 microplots in total, see Figure 3.18 for an illustration. We follow the conclusions of the joint plant and stem detection experiments in Section 6.7 and use FCN-UAV-STEM in its two task-specific decoder variant, as the performance is better for both stem detection and plant classification. Therefore, we labeled the ground truth for these experiments considering the pixel-wise labeling of crop plants, weeds, and background. Besides, we labeled the stems for the crop plants. As test data, we use the remaining 105 microplots. For these plots, we manually counted the number of plants per plot in the UAV images. As a baseline, we furthermore evaluate our FCN-UAV approach that solely provides the pixel-wise classification into the classes crop, weed, and background. To count the plants, we perform a subsequent connected component analysis on the predicted class label mask by only considering the crop class.

To analyze the plant counting performance, for each micro plot m, we compare the predicted number of plants c_p with the number of plants coming from the ground truth c_{gt}. We calculate the absolute error over per micro plot by

$$AE_m = \sum_{M} |c_p - c_{gt}| \, . \tag{6.8}$$

We then measure the performance based on the individual measurement days. We analyze the same microplots 20 days after seeding (DAS-20), 34 days after seeding (DAS-34), and 52 days after seeding (DAS-52), via the mean absolute error across as

$$MAE_{DAS} = \sum_{m=1}^{35} AE_m \quad \text{with } DAS = \{20, 34, 54\}, \tag{6.9}$$

Table 6.25: Plant counting performance of FCN-UAV-STEM and FCN-UAV on the GOETT-UAV-19 dataset.

Approach	DAS-20			DAS-34			DAS-52		
	MAE	MAE [%]	MAXE	MAE	MAE [%]	MAXE	MAE	MAE [%]	MAXE
Training: GOETT-UAV-19 15 plots Deployment: 105 plots									
FCN-UAV-STEM	9.8	4.8	14	7.6	3.7	12	9.6	4.7	15
FCN-UAV	19.0	9.3	37	24.4	11.9	52	35.8	17.5	64

and in addition to that, we report the maximum absolute error per measurement day MAXE, which is found for a single micro plot.

Table 6.25 summarizes the obtained counting performance achieved by FCN-UAV-STEM and FCN-UAV. These results demonstrate the superior counting performance of our proposed FCN-UAV-STEM architecture compared to the baseline. Across all measurement days, it achieves an average MAE of 9 plants per plot, which reflects a counting error of 4.4 %. 15 plants give the maximum counting error for a single plot. These results are comparable to human performance.

The FCN-SEQ approach achieves a comparably low performance with an MAE of around 24 plants per plot. In the worst case, it produces a counting error of 64 plants for a single plot. Thus, the performance is not suitable for crop counting. The reason for the larger error is depicted in Figure 6.26. While FCN-UAV-STEM can properly locate the stems of individual crop plants, the connected component assumption for the postprocessing of the FCN-UAV result does not hold, as over-segmented crop plant objects lead to too many counts per plot. The over-segmentation is mostly caused by the straw that fragments individual plants in image-space and mistakes in the pixel-wise classification. FCN-UAV-STEM is robust to both of these effects and thus provides a stable estimate for the stand count, even when the pixel-wise classification into crop and weed is not always correct.

We conclude that our FCN-UAV-STEM approach is suitable to perform UAV-based automated crop counting in harsh conditions.

6.10.1 Conclusions for the UAV-based Crop-Weed Classification

We draw the following conclusions from the presented results:

First, for high-resolution UAV images, fully convolutional neural networks provide better crop-weed classification performance under similar as well as under changing field conditions compared to the random forest-based approaches. One reason for the better performance is that fully convolutional neural networks can better deal with the separation of vegetation and soil. Especially under changing field conditions, the threshold-based vegetation classification reveals unreliable. For a high-quality performance, it has to be adapted manually.

Second, for low-resolution UAV images, the FCN-UAV approach performs better

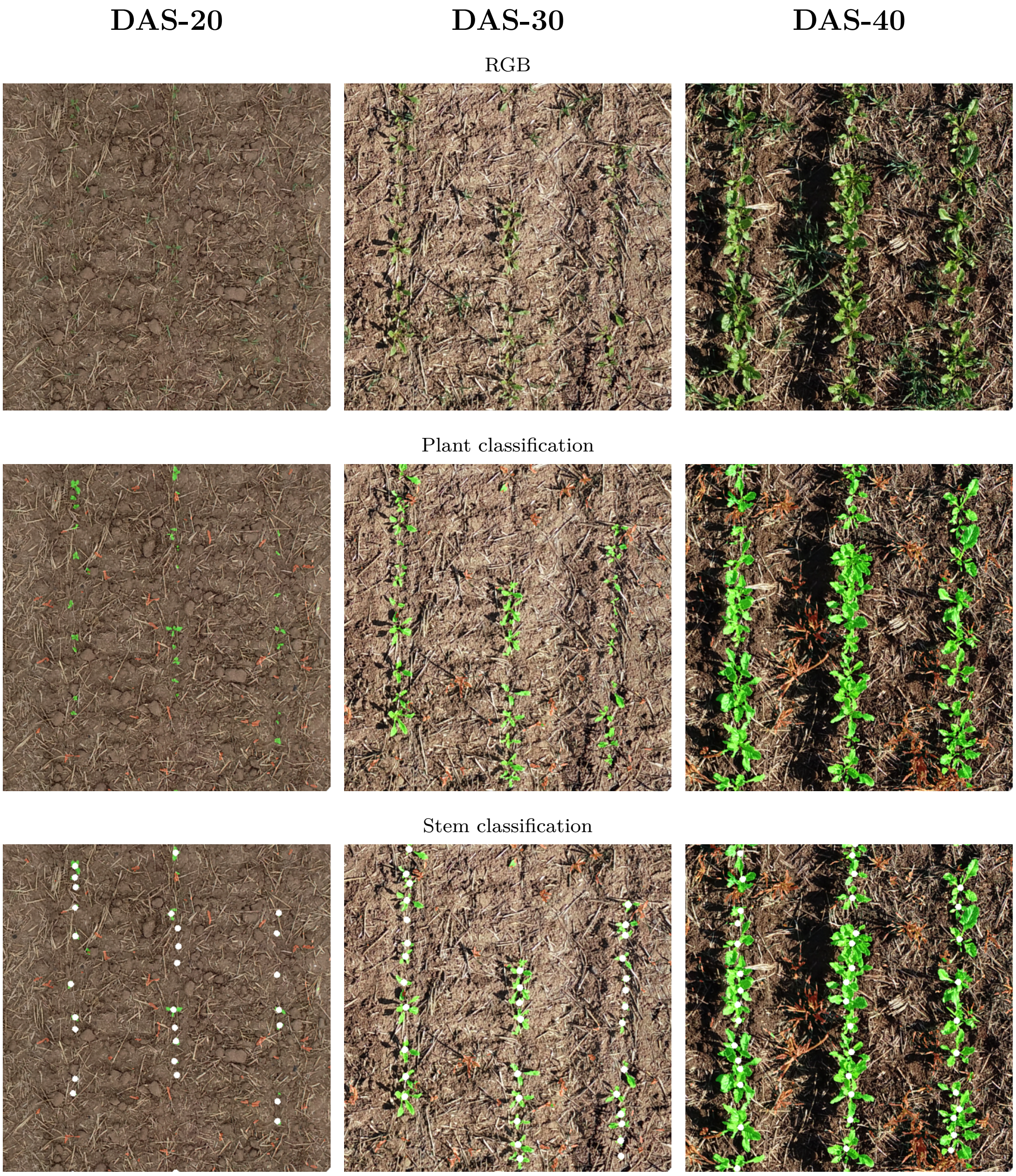


Table 6.26: Zoomed view of microplots and corresponding crop-weed classification and stem detection results.

than the RF-UAV approach. As in the case of the high-resolution images, the RF-UAV approach produces more missclassifications. First, in the case of weeds that are located close to the crop row. Second, for weeds that overlap with the crops. The performance of the random forest approach is limited for lower ground resolution.

Third, the explicit modeling of geometric features substantially helps to bridge the performance loss when deploying classifiers in new and unseen field environments. Thus, we recommend exploiting the crop row information, preferably under new and changing field conditions.

Fourth, regarding the UAV-based multi-species classification, the FCN-UAV approach performs better than the RF-UAV approach. Fully convolutional neural networks can extract more descriptive features to distinguish different weed species.

Fifth, the FCN-UAV-STEM approach is suitable to perform UAV-based automated crop counting, even in harsh conditions with high weed pressure and overlapping crop plants. The stem detection serves as a good method to extract the plant count from the data for several growth stages.

6.11 Runtime Performance for In-Field Treatments

A classifier suited for online plant classification must work a minimum required runtime to ensure a desired throughput of the robotic weed control system. Within Section 6.1, we define the minimum required runtime for our setup to be around 5 Hz. The last experiment in our experimental evaluation is designed to support the claim that our proposed approaches run fast enough to provide suitable results for the online operation of a robotic weed control system. We evaluate the runtime on the computer that is installed on the field robot (Intel i7 CPU, Nvidia GeForce GTX-2080 GPU).

We implemented all parts of our random-forest based classification pipeline in C++. We implemented most of the components in the pipeline using CUDA to exploit the parallel processing capabilities of the GPU. For the random forest and some basic functionality, we use the OpenCV library [2]. For the Markov random field smoothing, we use the implementation provided by Felzenszwalb *et al.* [30]. We implemented all parts concerning the pipeline for the fully convolutional neural networks in Python. For the design and training of the networks, we use Keras [23]. For the inference, we use Tensorflow [2].

Note that we do not claim that our implementations are fully optimized for a fast runtime. We deploy the random forest on the CPU running the inference on eight trees in parallel, each tree on a separate core. For the fully convolutional neural network-based approaches, we neither optimize the inference graph using the TensorRT library for faster inference, nor do we use quantized inference. Thus, there is still potential to speed up the runtime of our algorithms.

For all experiments in this section, we deploy the preprocessing on the GPU. For the random forest-based approaches, we use our CUDA implementation, whereas for

Table 6.27: Runtime analysis of the random forest-based and the end-to-end fully convolutional neural network-based plant classification approaches. For the random forest-based pipeline, we present the runtime for the key processing steps. We report the runtime for the inference on the BONN-CW-16 dataset using RGB+NIR images of size 512 × 384 as the input.

Function	**Device**	**Min** [ms]	**Std** [ms]	**Max** [ms]
Random forest approaches				
Preprocessing	GPU	18	3	20
Vegetation Classification	GPU	15	1	18
Object based RF-OBJ				
Feature extraction	GPU	22	3	31
Classification	GPU	4	2	15
Overall		69		97
Keypoint based RF-KP				
Feature extraction	GPU	89	25	213
Classification	CPU	85	22	149
MRF smoothing	CPU	356	78	566
Overall		563		966
Semi-supervised RF-GC				
Geometric classification	CPU	8	1	15
Geometric classifier update	CPU	4	2	6
Fully convolutional neural network approaches				
FCN	GPU	30	1	32
FCN-STEM	GPU	53	1	54
FCN-SEQ	GPU	83	1	85
FCN-SEQ-STEM	GPU	154	1	156

the fully convolutional neural networks, we perform the preprocessing using the native TensorFlow functionality. For all experiments in this section, we perform the classification on the BONN-CW-16 dataset to record the runtime for the respective pipelines. We use RGB+NIR images of size 512 × 384 as the input.

We exploit the GPU using our CUDA implementation for the vegetation classification, which we describe in Section 4.3.1, as well as for the extraction of the keypoint-based and object-based features, which we describe in Section 4.3.3. As the keypoint and object features are computed only for image regions that correspond to vegetation, the total runtime of the RF-KP and RF-OBJ approach depends on the amount of classified vegetation for each image.

Table 6.27 illustrates an overview of the execution time of different parts of our classification system for the RF-KP and the RF-OBJ approach. The average execution time to extract the handcrafted features is around 90 ms for the keypoints and 22 ms for the objects. Furthermore, we use the GPU for our preprocessing step to obtain normalized intensities for each input channel. The execution time for this task is around five times faster compared to the CPU implementation, see Table 6.27. The most time-consuming part is currently the MRF smoothing of the keypoint-based approach, which runs on the CPU. Generally, we need many more keypoints than objects to cover the vegetation of an image. Thus, the object-based classification is substantially faster. The RF-OBJ approach requires, on average, around 70 ms for the inference on a single image, whereas the keypoint approach RF-KP needs around 560 ms and in the worst case almost a second. Thus, on its own, it is too slow for being applied for on-field weed control. The geometric classifier of the RF-UAV approach contributes on average a negligible amount with around 12 ms of extra runtime to the pipeline. The cascaded RF-CAS approach combining RF-OBJ and RF-KP provides an average runtime of around 110 ms. However, in the worst case, the total runtime can sum up to the one of RF-OBJ plus RF-KP, which is too slow. To ensure a fast enough runtime, we can constrain the number of keypoints and do not use the MRF smoothing.

We also report the inference time for the fully convolutional neural networks in Table 6.27. Note that the reported runtime covers the entire duration for inferring the classification result for a single image, also including the preprocessing. We see that with an increasing model capacity, the runtime slows down. Our FCN approach provides the fastest runtime with around 30 ms, which corresponds to the processing of 33 images per second. Even the complex FCN-SEQ-STEM model using two task-specific decoders and the sequential module achieves a runtime of around 154 ms per image, i.e., 6.5 images per second. In the course of implementing our FCN approach on a small mobile robot for selectively spraying weeds in rail tracks, we compiled the FCN architecture using the TensorRT library for the NVIDIA Jetson Xavier. Here, we achieved a runtime of about 50 ms, which corresponds to predicting 20 images per second.

We conclude that all proposed approaches in this thesis provide an average runtime of >5 Hz, enabling agricultural robots to perform on-field weed control. Here, we exclude the keypoint-based approach, as it is not deployed as a standalone for the UGV-based applications. Fully convolutional neural networks are beneficial compared to the random forest-based approaches. Fully convolutional neural networks provide both a faster runtime and a generally better classification performance. Furthermore, nowadays, they are way easier to implement, as several optimized open-source libraries exist that handle most of the low-level operations exploiting dedicated hardware components.

Chapter 7

Related Work

For more than two decades, estimating semantic information from sensor data has been a highly relevant topic in robotics and computer vision [107]. Various approaches have been proposed to analyze the environment around a robot in indoor as well as in outdoor environments [9, 11, 68, 74, 75, 139, 148, 157], e.g., for optimizing mapping, traversability analysis [11, 139, 157], navigation [68], object detection [74], pedestrian detection [75], autonomous driving [9], face detection [148], and for many other applications. In the past, a variety of classification techniques such as Boosting methods [34], support vector machines [13], or random forests [15] have been applied. We refer to these kinds of methods as traditional machine-learning approaches. Within the last ten years, however, deep learning has revolutionized the semantic interpretation of image or laser range data in a large number of domains, including precision farming.

In the field of precision farming and agricultural robotics, semantic interpretation of the sensor data plays a major role and is a key driver for several directions for development such as autonomous navigation [33, 116, 155], task planning [4], site-specific and selective treatments [82, 84], autonomous harvesting [72, 73, 130], and many more [7, 8, 110]. To infer actionable data from the field status and to enable robots to perform targeted weed control, selective sampling, manipulation, and harvesting, we need reliable and robust working classification systems that can detect the desired targets in the field. Shamshiri *et al.* [133] provide a comprehensive overview of recent achievements in agricultural robotics, specifically for autonomous weed control, field scouting, and harvesting. They conclude that the research and development in these fields have made significant progress. On the other hand, prototype robots for weeding and harvesting are not close to being able to compete with the human operator, yet. Thus, robotics has not reached a commercial scale for agricultural applications for now. As major challenges, they identify robust perception, task planning algorithms, and optimization of sensors. Their conclusion is in line with the one of Bechar and Vigneault [7, 8]. They conclude that information-acquisition systems in agriculture, including sensors, fusion algorithms, and data analysis, need to be adjusted to the dynamic conditions of unstructured agricultural environments. Agricultural robotic systems for plant cultivation operate under unstructured agricultural environments without compromising the productivity and quality of work compared to currently employed methods. Across a variety of applications, common problems for successful commercialization are poor detection performance, inappropriate decision-making, and low action success rate in unstructured, dynamic environments.

In this chapter, we present the work related to the developments in this thesis. We divide this chapter into six sections in which we deal with individual aspects of the structure of this work. In Section 7.1, we present different works that use traditional machine techniques. in Section 7.2, we present works that use deep learning. Section 7.3 handles approaches that particularity aim to detect plant stems, enabling robots for precise mechanical intervention. Section 7.4 deals with related work, which is specifically related to data analysis of UAV images. Note that the papers in this section use traditional and modern machine-learning approaches. Finally, our Section 7.6 deals with works that are dedicated to solving specific challenges such as the generalization capabilities of plant classifiers to new and changing field conditions. Here, we present works that aim to reduce the effort of the laborious labeling of the data.

7.1 Traditional Classification Systems Based on Handcrafted Features

Traditional approaches rely on machine-learning algorithms based on handcrafted features encoding the image content utilizing statistical, color, shape, and texture descriptors. These handcrafted features are usually adjusted to the crop plants and weeds that are present in the specific dataset or a particular application. Thus, relying on handcrafted features usually involves the tweaking of parameters to adapt them to a different situation. A classifier using these features will always be limited by the employed features and by the information extracted by these features. Therefore, much of the research has focused on the development of more complicated non-linear classifiers, such as support vector machines or random forests, to overcome the limitations of the employed features.

In this field, several approaches, such as [50, 53, 103], are often based on a supervised machine-learning classification system, which needs to be trained on domain-specific data in order to predict the desired output during the operational phase. Mostly, the concept of such systems is given by a two-stage classification system, where the first step is to separate the vegetation, and the second step is to analyze the vegetation parts further and distinguish them into crops and weeds or multiple species of weeds. Haug *et al.* [50] propose a method to distinguish carrot plants and weeds in RGB and near-infrared images that are acquired with the same camera system that we also employ in this work, see Section 3.1.1. They obtain an average accuracy of 94 % under similar field conditions on a dataset containing 70 images. Hemming and Rath [53] propose a vision-based system that distinguishes carrots, cabbage, and weeds using a fuzzy logic classifier. For leaf classification based on RGB images, they perform a pre-segmentation into individual plants to extract shape and color features per segment. They evaluate their approach in open field experiments and achieve classification accuracies of 72 % up to 88 %, but report that pre-segmentation of plants entails problems and embodies a limiting factor of their approach. This conclusion coincides with ours that the vegetation classification on pure RGB data sometimes does not provide sufficient results in terms of the vegetation classification. Nieuwenhuizen [103] presents an approach in his thesis on the automated detection and control of volunteer potato plants in sugar beet fields based on a Bayesian classifier and an artificial neural network classifier. Both classifiers are fed with features encoding mostly color information. He observes substantial performance differences when deploying the trained classifier in different field environments, where the illumination conditions have changed. He concludes that the chosen handcrafted features do not generalize well to previously changing field conditions. He furthermore observes that the neural network classifier generally performs better than the Bayesian classifier.

Also, our random forest-based classification systems, which we describe in Chapter 4, follow a two-step approach. They first recognize the vegetation and then divide the vegetation into the desired classes. We use random forests for the classification as they provide comparatively robust results and use a variety of different color, shape, and texture features for the classification. However, to achieve both, a fast execution time as well as the ability to deal with overlapping plants, we combine both approaches and propose our cascaded RF-CAS approach in Section 4.3.6.

In the context of agricultural applications, several vision-based crop and weed detection approaches for specific plants have been proposed, and innovative solutions have been developed for in-field treatments. Müter *et al.* [101] propose a mechanical approach that focuses on the removal of weeds through the design and control of a mechanism for intra-row weeding. McCool *et al.* [91] propose a study about the efficiency of different mechanical tools for robotic weeding. They consider tilling below the surface with arrow hoes, above-surface tilling by tines, and weed cutting. They find that the above-surface tilling by tines is most effective for a variety of weed species, but also point out the importance of early intervention. Lehnert *et al.* [73] demonstrate their robotic harvester that can autonomously harvest sweet pepper in protected cropping environments. In their paper, they explain the design and functionality of the harvester, including scanning, detection, grasping, and the picking of the sweet pepper crops. Pretto *et al.* [120] summarizes the work and developments of the EC-funded project Flourish, in the context of which this work is carried out. For autonomous weed control, we use a collaborative approach of UAVs and UGVs, which first detect weeds in the field and then control them with different treatment solutions, i.e., by selective spraying or precise mechanical stamping.

Also, concerning robotic harvesting systems, the robots must be equipped with a fruit recognition system that works under practical conditions. McCool *et al.* [93] present a crop detection system applied to the task of field sweet pepper detection. The paper deals with the detection of red and green sweet pepper on a green background formed by leaves. The authors can detect 70 % of highly occluded crops under practical conditions. They propose a two-step processing pipeline. First, they segment the crop using a conditional random field based on visual texture features. Second, they analyze the resulting pixel-wise probability map for the sweet pepper detection by applying a Laplacian of Gaussian multi-scale blob detector. Sa *et al.* [130] present a 3D visual detection method for detecting peduncles of sweet peppers in the field. They exploit both color and geometry information acquired from an RGB-D sensor and utilize a supervised learning approach for the peduncle detection task. They achieve high classification performance for red sweet pepper examples on a background consisting of green leaves. However, for green sweet pepper examples, the performance decreases due to the similar color features concerning the background.

In terms of multispectral data analysis, several works have been published. Borregaard *et al.* [12] perform crop versus weed classification using narrowband reflectance at 694 nm and 970 nm. Feyaerts *et al.* [31] conduct a multispectral machine vision study to design an online weed detection system for selective spraying. They collect multispectral images using six channels with different wavelengths, i.e., 441 nm, 446 nm, 459 nm, 883 nm, 924 nm, and 988 nm, in the field. They report crop versus weed classification rates of 80 % for sugar beet plants and 91 % for weeds. Strothmann *et al.* [140] use a multi-wavelength line scanner for crop and weed classification using a Bayesian approach. To adapt their classifier, they label a small amount of data covering the actual feature distribution and retrain their classifier.

Other researchers have investigated the use of texture computed from grayscale and color images to identify plant species. Shearer *et al.* [134] used gray level co-occurrence matrices in the hue-saturation-value (HSV) color space. Related to that, Burks *et al.* [17] evaluated the color texture classification of different weed species using a neural network classifier. Both works report that using statistical parameters extracted from co-occurrence matrices provide high discriminative power to identify or separate plants but accentuate that more research is needed for testing under uncontrolled field conditions. Latte *et al.* [70] use features based on color space and gray-level co-occurrence matrices to classify crop field images containing eight different types of crop. They apply an artificial neural network classifier and achieve an average classification accuracy of 84 % when using both the HSV based and gray-level co-occurrence matrices based features. Their results show that using statistical moments of the HSV distribution improves the overall classification performance. McCool *et al.* [90] develop an approach to automate the process of vegetation cover estimation. In their paper, they address the problem of distinguishing grasses and weeds in RGB images, where the weeds are located with a field of grasses, i.e., solving a “green-on-green” problem. First, they model the distribution of color using a multivariate Gaussian based on the Lab, Luv, and HSV color spaces to separate the vegetation from the background. In a second step, they classify each vegetation pixel regarding its class affiliation being grass weed or herb. This classification step is performed by template matching using local binary pattern features.

Several works have been conducted in the context of leaf image classification and segmentation [18, 67, 152]. In work done by Wang *et al.* [152], leaf images are segmented using morphological operators, and shape features are extracted and used in a moving center hypersphere classifier to infer plant species. Kumar *et al.* [67] start from segmented images of leaf using a binary classifier on global image signatures as a validity test and curvature features compared with a given database to extract the best match. To cover a variety of leaf shapes, also deformable leaf models and morphology descriptors have been exploited by Cerruti *et al.* [18]. Elhariri *et al.* [29] compared a random forest classifier and a linear discriminant analysis-based approach in their study to classify 15 plant species through leaf images. They exploit HSV color space of leaf images as well as gray level co-occurrence matrices to extract shape, color and vein features. Hall *et al.* [46] conduct a study on features for leaf classification. They compare classification performance on different feature types like typical handcrafted and convolutional neural network (ConvNet) features using a random forest classifier. They evaluate the robustness of those features under simulated varying conditions on the public Flavia leaf dataset. They report an average accuracy of around 97 % by combining traditional and ConvNet features and conclude that the combination of handcrafted and ConvNet features adds robustness to varying conditions in the classification process.

Tellache *et al.* [143] present a vision-based approach for selective weed spraying. They capture images inclined downwards concerning the horizontal plane of field scenes and subdivide them into grid cells. For each cell, a decision is made based on structural and area features using Bayesian decision theory. A further cell-based approach by Aitkenhead *et al.* [3] fragments images in a top-down fashion containing seedlings of crop and weeds into 16 cells and classify each of them using a self-organized neural network. They attain a classification performance close to 80 %, but as in the case of Tellaeche *et al.* [143] at a comparably low resolution of the cells. In contrast, we provide labels for the full image resolution to allow a high precision treatment in object space. In contrast, all our proposed approaches in this thesis provide labels for the full image resolution at the pixel-level to allow a high precision treatment in object space.

Other works such as Gai *et al.* [36] and Weiss and Biber [153] exploit 3D information obtained from sensors such as laser scanners and depth cameras to separate crops from the background soil for localization and mapping applications. Gai *et al.* [36] propose a system to recognize and localize broccoli and lettuce plants based on a combination of 2D and 3D information. They use a Kinect V2 sensor for data acquisition in real fields, observing different growth stages of the plants. They train different classifiers, such as support vector machines, random forests, and logistic regression, using different handcrafted features describing color, texture, and morphology in 2D images as well as in 3D morphology. As a preliminary step, they separate the vegetation from the soil based on hight difference by exploiting the 3D information. They state that the major challenges of their proposed approach are the wrong segmentation of the vegetation situation with high weed pressure and poor plant segmentation for small plants at the early growth stage. Weiss and Bieber [153] present in their article an approach for in-field classification and mapping of maize plants with mobile robots using 3D LIDAR information. They show that they can reliably detect the maize plants and distinguish them from the ground, despite the use of a comparably low-resolution FX6 LIDAR sensor by Nippon. Nevertheless, both approaches are based on the segmentation of plants utilizing 3D information, or precisely, the plant height. However, this approach may result in small or generally flat plants that cannot be identified robustly because the signal is not sufficient for a too noisy environment.

Therefore, we focus on the detection of plants using high-resolution RGB or RGB+NIR image data. Due to the high soil resolution, we can detect even the smallest plants or even seedlings with a size of $0.2\,\mathrm{cm}^2$ or more. This is essential for effective weed control, as weeds must be detected and eliminated as early as possible during the vegetation period.

7.2 Modern Classification Systems Based on Deep Learning

The advent of end-to-end trainable convolutional neural networks [66] spurred interest in end-to-end learnable crop-weed classification pipelines to overcome the earlier described limitations of handcrafted pipelines since they allow to learn feature representations directly from the training data using backpropagation [128]. Such a richer feature representation aggregated over multiple layers of convolutions, pooling operations, and non-linearities enables the convolutional neural networks to get away with simple linear classifiers on top of these more complex features compared to those mentioned above simpler handcrafted features.

A considerable number of publications treat the classification of plants and herbs as an image classification problem, i.e., the classifier predicts a predefined number of classes for the entire input image. Lee *et al.* [71] study the performance of convolutional neural networks for the classification of 44 different plant species based on the AlexNet architecture proposed in Krizhevsky *et al.* [66] for a leaf dataset. This dataset is named MalayaKew Leaf Dataset collected at the Royal Botanic Gardens in England. The dataset contains images from pre-segmented leaves. They train the classifier on 2,300 leaf images and report a classification accuracy of 99 % achieved on 530 test images. Furthermore, they investigate a visualization technique for the importance of features based on deconvolutional networks. Their results show that the network mostly exploits the venation structure of the leaves to identify different plant species. Olsen *et al.* [106] propose the DeepWeeds dataset. It is a multi-class image dataset of weed species from the Australian rangelands. It consists of 17,509 labeled images concerning eight different weed species. The images are labeled according to the presence of one or more specific weeds. Thus, the data is labeled for being treated as an image classification problem. The classifier should output a probability for each weed species being in an image. Along with the dataset, the authors present two convolutional neural network baselines following the Inception-V3 and ResNet-50 architectures. They initialized the models with weights that were trained on the ImageNet dataset and fine-tuned the models with samples from their proposed DeepWeeds dataset. They report classification accuracies in the order of 95 % for the multi-class problem. Teimouri *et al.* [142] propose an image classification approach based on the Inception-V3 architecture for counting leaves of 18 different weed species. They present the classifier cropped images from different weeds that are typical in Denmark. They consider nine classes ranging from one leaf to nine leaves. They train the network on almost 12,000 RGB images and report a classification accuracy of around 70 % for the leaf counting on a test dataset containing around 2,500 RGB images. However, their evaluation of the confusion matrix suggests that most of the errors lie in the range of one leaf concerning the ground truth.

A limiting factor of image classification is that the classification decision applies to the entire image. Therefore, the spatial resolution is always defined by the image content. To address this problem, convolutional neural networks are often applied in a pixel-wise fashion operating on image patches provided by a sliding window approach. Using this principle, Potena *et al.* [117] use a cascade of convolutional neural networks for crop-weed classification, where the first convolutional neural network detects vegetation, and only then the vegetation pixels are classified by a deeper crop-weed convolutional neural network. These pixel-wise approaches operating on small patches extracted from the image can only use very local information present inside the patch. This limits the receptive field of the convolutions and, therefore, also the amount of context incorporated into the classifier. All of our proposed fully convolutional neural network architectures use the whole image instead and provide a pixel-wise classification. Therefore, our approaches use potentially information from the whole image in higher layers. Fully convolutional networks [80] directly estimate a pixel-wise segmentation of the complete image and can, therefore, use information from the whole image. The encoder-decoder architecture of SegNet [5] is nowadays a common building block of semantic segmentation approaches [108, 126].

In the past three years, other papers have been published that treat plant recognition as a pixel-wise classification. The work by Milioto *et al.* [97] combines an effective end-to-end semantic segmentation also based on a fully convolutional network architecture with plant features, which are comprised of low-level image features, like a vegetation index. They also show that the network can be fine-tuned to novel data using only very few labeled images. Cechlinski *et al.* [21] propose an online crop-weed classification system that works at 10 frames per second with a Raspberry Pi minicomputer. Their approach performs a pixel-wise classification into the classes crop, weed, and soil. They exploit the MobileNet-V2 [132] architecture, which is designed for high-speed inference on mobile devices and propose a further architectural extension to it based on DenseNet architectures [55, 58].

Yu *et al.* [158] propose an approach for weed detection in ryegrass based on convolutional neural networks to enable mobile robots for selective spot-spraying. The authors treat the problem as an object-detection problem that predicts bounding boxes around the present weeds in the data and report F1-scores for weed detection of around 93 %. They also train the network to predict four different weed species. Here, they report an average recall of 93 % at a precision of 70 % under similar field conditions. They do not explicitly evaluate changing field conditions. McCool *et al.* [92] propose an approach to obtain a mixture of lightweight convolutional neural networks that are suitable to be deployed on small robotic platforms. The goal of the networks is to provide an appropriate crop-weed classification performance that enable the robot for selective and plant-specific in-field treatments. Common state-of-the-art architectures, such as Inception-V3 [141], are too large and thus too slow for being deployed for online operations in the field. Therefore, the authors present a three-step approach to compress the model architecture such that it is suitable for online processing. First, they fine-tune a very deep convolutional neural network based on the Inception-V3 architecture on the training data. Second, they apply model compression by training the lightweight "student" networks using the output of the large "teacher" network. Third, they ensemble several trained lightweight models and obtain a classification accuracy of around 90 % for the weed detection task. This performance is 4 % less compared to the one obtained by the Inception-V3 model, but also 4 % higher compared to a traditional approach.

7.3 Plant Stem Detection for Precise Intervention

Several works have focused on identifying the stem locations of the plants. Most of these approaches are also based on handcrafted heuristics targeted towards specific applications. Kiani *et al.* [62] use handcrafted shape features selected through a discriminant analysis to differentiate corn plants from weeds and identify stem positions of the plants as the centroid of the detected vegetation. This leads to sub-optimal results, mainly when the plant shapes are not symmetric or multiple plants are overlapping. Midtiby *et al.* [95] present an approach tailored to sugar beet plants by detecting individual leaves and using the contours of the leaves for finding the stem locations. However, such approaches usually fail to locate the stems in the presence of occluded leaves or overlapping plants. Langer *et al.* [69] propose a two-step approach for geometric stem detection. First, they detect the vegetation using a threshold-based approach based on vegetation indexes in RGB and near-infrared images. Then, an initial leaf detection is performed using convexity defects in the convex hull of connected components in the binary mask that encodes the detected vegetation. The final stem detection is then obtained from the intersection of the estimated main axis of the leaves.

Moving in the direction of a data-driven machine-learning approaches, Haug *et al.* [49] propose a system to detect plant stems using keypoint-based random forests. They use a sliding window-based classifier to predict stem regions by using several handcrafted geometric and statistical features. Their evaluation shows that the approach often misses several stems of overlapping plants or generates false positives for leaf areas that locally appear to be stem regions. Krämer *et al.* [65] aim at addressing this issue by increasing the field of view of the classifier using fully convolutional networks. The goal of their work is to identify plant stems over a temporal period allowing them to use stem locations as landmarks for robot localization in the field.

Our work overcomes many of the limitations by taking a holistic approach by jointly detecting stems and estimating a pixel-wise segmentation of the plants based on FCNs. Our network shares the encoded features for classifying the stem regions as well as for the pixel-wise classification using an encoder network along with two, task-specific decoder networks. Moreover, we explicitly distinguish crop and dicotyl weeds stems since it enables plant-specific treatment, for example, fertilizing a crop or destroying a weed mechanically.

Another approach, which does not perform stem detection, but provides a multi-task network benefit is the one by Pound *et al.* [118]. They present a multi-task classifier based on stacked encoder-decoder structured, fully convolutional neural networks, as proposed by Newell *et al.* [102]. They analyze high-resolution images containing wheat and perform a pixel-wise classification to segment the spikelets and detect the spikes jointly. They furthermore show that it is possible to add a third task to the network, i.e., to predict for a shown image if the plants are awned or not. Based on a dataset containing 520 images, they train and test their network and report classification accuracies of 96 % for spikes and 99 % for spikelets.

7.4 UAV-Based Field Monitoring

UAVs equipped with different sensors serve as an excellent platform to obtain fast and detailed information on arable field environments [150]. Monitoring crop height, canopy cover, leaf area, nitrogen levels, or different vegetation indices over time can help to automate data interpretation and thus to improve crop management, see [37, 61, 115, 146]. Geipel *et al.* [37] as well as Khanna *et al.* [61] focus in their work on the estimation of crop height using UAV imagery. Both these works apply a bundle adjustment procedure to compute a terrain model and perform a vegetation segmentation in order to estimate the crop height based on the obtained 3D information. Tokekar and Hook [146] introduce a concept for a collaboration of a UGV and a UAV with the goal to measure nitrogen levels of the soil across a farm. They use UAVs for the measurements and UGVs for the transport of the UAVs due to its limited energy budget.

Several works have been conducted in the context of vegetation detection by using RGB as well as multispectral imagery of agricultural fields [42, 48, 147]. Hamuda *et al.* [48] present a comprehensive study about plant segmentation in field images by using threshold-based methods and learning-based approaches. Torres Sanchez *et al.* [147] investigate an automatic thresholding method based on the normalized difference vegetation index and the excess green index in order to separate the vegetation from the background. They achieve an accuracy of 90-100 % for the vegetation detection based on their approach. In contrast, Guo *et al.* [42] apply a learning approach based on decision trees for vegetation detection. They use spectral features for the classification exploiting different color spaces based on RGB images. We use a threshold-based approach based on the excess green index and normalized difference vegetation index in order to separate the vegetation from the background, i.e., mostly soil. Fuentes-Pacheco *et al.* [35] propose a self-designed encoder-decoder structured, fully convolutional neural network for pixel-wise crop classification in UAV images. They classify fig plants in RGB images of a crop grown under difficult circumstances of complex lighting conditions. They train the classifier on a small amount of manually labeled images considering the classes fig and non-fig. They report a mean pixel-wise classification accuracy of around 94 %. As the training and test data are sampled from the same dataset, we consider this as the performance under similar field conditions. Furthermore, they perform the classification by a threshold-based approach using different RGB-based vegetation indexes. They observe that the threshold-based classification provides acceptable performance but is clearly surpassed by the convolutional neural network approach.

The next level of data interpretation is the classification of the detected vegetation by separating it into the classes crop and weed. Several approaches have been proposed in this context. Peña *et al.* [111] introduced a method for the computation of weed maps in maize fields based on multispectral imagery. They extract super-pixels based on spatial and spectral characteristics, perform a segmentation of the vegetation, and detect crop rows in the images. Finally, they use the information about the detected crop rows to distinguish crops and weeds. In a follow-up work by Peña *et al.* [112], they evaluate a similar approach according to Peña *et al.* [111] for different flight altitudes and achieve the best performance, i.e., around 90 % overall accuracy for crop/weed classification using images captured at an altitude of around 40 m with a spatial resolution of 15 $\frac{mm}{px}$. Furthermore, they conclude that using additional near-infrared information leads to better results for vegetation detection. Similarly, Montalvo *et al.* [99] perform a crop row detection for high weed pressure based on a Hough transform and use of prior knowledge about the crop rows location within the images.

Also, machine-learning techniques have been applied to classify crops and weeds, in UAV imagery of plantation [41, 57, 113, 114]. Perez-Ortiz *et al.* [113] propose a weed detection system based on the classification of image patches into the values crop, weed, and soil. They use pixel intensities of multispectral images and geometric information about crop rows in order to build features for the classification. They evaluate different machine-learning algorithms and achieve overall accuracies of 75-87 % for the classification. Perez-Ortiz *et al.* [114] use a support vector machine classifier for crop/weed detection in RGB images of sunflower and maize fields. They present a method for both inter-row and intra-row weed detection by exploiting statistics of pixel intensities, textures, shape, and geometrical information as features. Guerrero *et al.* [41] propose a method for weed detection in images of a maize field, which allows identifying the weeds after its visual appearance changed in image space due to rainfall, a dry spell, or herbicide treatment. Garcia *et al.* [57] conduct a study on separating sugar beets and thistle based on multispectral images with a comparably large number of narrow bands. They applied a partial least squares discriminant analysis for the classification and achieved a recall of 84 % for beet and 93 % for thistle by using four narrow bands at 521 nm, 570 nm, 610 nm, and 658 nm for the feature extraction. Another noteworthy approach is the one by Mortensen *et al.* [100]. They apply a deep convolutional neural network for classifying different types of crops to estimate individual biomass amounts.

They use RGB images of field plots captured at 3 m above the soil, and report an overall accuracy of 80 % evaluated on a per-pixel basis.

Various studies have also investigated the use of multispectral cameras on UAVs for plant classification [59, 129, 131, 159]. Sa *et al.* [129] propose to use the SegNet [5] architecture for multispectral crop-weed classification from a UAV. They acquire narrow-band images in the near-infrared (790 nm) and the red (660 nm) spectrum and aim for classifying crop plants and weeds using an onboard computer mounted on the UAV. They obtain classification accuracies of around 85 % under similar field conditions. However, under changing field conditions, the classification fails. They conclude that more and diverse training data is required to train a proper model being robust under changing field conditions. In their follow-up paper, Sa *et al.* [131] provide the WeedMap framework. This article deals with UAV-based crop-weed classification through the analysis of multispectral images. The two differences to the previous paper are, that no online detection is required in the field and that the image classification is not performed on single images of the camera, but a previously processed orthomosaic of the entire field. This preprocessing of the data allows a spectral and geometric correction of the image data and thus increases the quality of the classifier's input. They use the same SegNet-based architecture as in Sa *et al.* [129], but this time feed the classifier with five input channels, i.e., red, green, blue, red-edge, near-infrared and four additional vegetation indexes. They achieve a classification accuracy of about 90 % with this setup on two sugar beet fields in Germany and Switzerland. They conclude that performance decreases drastically under changing field conditions. Another disadvantage of multispectral image data is the low spatial resolution of the camera compared to standard RGB cameras. This leads to the fact that small plants are often not recognized at practical flight altitudes.

In this thesis, we present two plant classification systems for analyzing UAV images. Our RF-UAV approach is based on random forests, and our FCN-UAV approach is based on fully convolutional neural networks. Both approaches can work on single images as well as on orthomosaics and can treat several classification problems classes, i.e., vegetation classification, plants, and weed classification, or can even detect different weed species. Besides, our RF-UAV approach explicitly exploits geometric features of the relative plant arrangement, thereby achieving better performance and generalization properties in new and changing field conditions.

Zhao *et al.* [159] propose a new method for rice lodging assessments based on a fully convolutional neural network exploiting the U-Net architecture [76]. They compare the classification performance for the lodging obtained by analyzing RGB with multispectral image data. They obtain an intersection over union score of around 93 % for the lodging areas with a slightly better performance with RGB data. However, they solely conduct studies where they train and test on examples coming from the same field environment. Bullock *et al.* [16] propose a binary image classification approach to detect grass weeds in maize crop using UAV imagery. Therefore, they divide the images into small patches and perform the binary classification for each patch. They analyze the effect of adding "context" on the classification performance through varying the image patch sizes to be analyzed. They report classification accuracies in the order of 96 % and suggest to select larger patch sizes to allow the network to learn features also considering more context, i.e., analyzing the surrounding location for potential weeds. Kampen *et al.* [59] use UAVs and multispectral images to classify trees and detect stress symptoms. Thus, even low-resolution multispectral cameras provide a sufficient number of pixels representing the tree. They analyze orthomosaics containing six different tree species and obtain a classification accuracy of around 96 %. They employ random forests as the classification model using spectral features. Furthermore, they perform disease detection and classify infested and non-infested trees with an accuracy of around 87 %. Their feature selection shows that the classification works best when using the red, red edge, and near-infrared reflectances.

Guo *et al.* [43] propose a two-step machine-learning, voting-based image processing method to detect and count the number of sorghum heads from high-resolution images captured by UAVs. The authors first train an ensemble of decision-tree classifiers on a pixel-wise basis using handcrafted features encoding color and texture. The goal of each decision tree is to provide a binary mask separating the image into sorghum and non-sorghum head regions. Second, they employ a voting process for all the segmented images from all decision trees to acquire the most reliably detected region of the sorghum heads, i.e., by filtering out outliers that do not have enough votes. Ghosal *et al.* [38] propose an active learning approach which is inspired by weakly-supervised deep learning approaches for sorghum head detection and counting from UAV-based images. They use a network architecture for object detection based on RetinaNet [79] and combine it with building blocks from the Resnet-34 [52] architecture. The key idea is to label only a few sorghum head examples, train the network, and deploy it on other images. Then, a human operator corrects errors made by the object detector. The new labels are then fed into the training process until the classifier obtains a satisfactory detection performance on a validation dataset. With our FCN-STEM approach, however, we also propose a framework based on fully convolutional neural networks for jointly classifying crop plants and weeds and counting.

7.5 Generalization Capabilities to New and Changing Field Condition

As both previous related work sections suggest, several vision-based methods have been proposed for plant classification. Typically, such approaches are based on supervised machine-learning techniques and report classification performances in the order of 70-95 % in terms of classification accuracy. However, most of the related works lack in the evaluation of several methods regarding the generalization capabilities to unseen situations, new fields, and changing field conditions. Precision-farming robots need to operate in different field environments regularly. A typical practical use case is that a plant classifier has been trained on data coming from one or more particular field environment, but is then deployed later in time or in another field where the visual appearance of the plants, weeds, and soil has notably changed. These changes can lead to different intensity distributions of the image data concerning the original training data. Both related works by Nieuwenhuizen [103] and Sa *et al.* [129] evaluate their classification systems under changing field conditions. In both works, the authors observe that the classifiers do not provide sufficient performance for the desired tasks.

We explicitly address this generalization gap in the performance of plant classification and propose novel approaches that exploit that a large number of crop plants are sown in rows. Sugar beet plants, for example, are arranged in crop rows and often share a similar lattice distance along the crop rows. Such geometric information is typically similar within and across fields and, thus, less dependent on the visual appearance of the plants. The following contributions aim at exploiting this geometric signal to improve the generalization capabilities of the plant classifiers.

For the random forest-based classification, we propose the RF-GC approach that integrates the geometric information into the visual plant classification system. We combine the visual and the geometric classifier to compute a joint classification of the crop plants and weeds and to achieve an online adaption of the visual classifier to match better with the actual distribution of the visual features. For the FCN-based approaches, we propose a novel way to exploit additional geometric prior information about the local arrangement of the plants in the field by analyzing image sequences that cover a local strip of the field surface and thus implicitly carry the information about the plant arrangement. We introduce a subnetwork, called sequential module, that analyzes visual features of consecutive images from a sequence and extracts spatio-temporal features that encode the field geometry. The integration of the sequential module into our FCN approach leads to our proposed novel FCN-SEQ and FCN-SEQ-STEM approaches.

7.6 Reducing the Required Labeling Effort

A further challenge for supervised classification approaches is the necessary amount of labeled data. The labeled data is typically obtained at a high cost. In recent years, several studies have focused on how to minimize the effort for the data labeling itself or on how to minimize the number of labeled data points to transfer a classifier for a new field domain.

The easiest way to adapt a classifier to new conditions is to retrain it on new labeled data from the new domain. This approach is called supervised transfer learning. Strothmann *et al.* [140] use a multi-wavelength line scanner for crop and weed classification using a Bayesian approach. To adapt their classifier, they label a small amount of data covering the actual feature distribution and retrain their classifier. Bosilj *et al.* [14] study in their article the retraining efforts that are required to transfer classifiers between different types of crop and different field conditions to obtain sufficient classification performance for crop plants and weeds. They use convolutional neural networks and propose a two-step process to label data for the targeted domain in order to adapt the classifier. First, they let the classifier predict a pixel-wise classification of an image in the targeted field environment. Second, they correct the prediction of the classifier instead of creating annotations from scratch for the target domain. Through this procedure, they save time for labeling effort and focus on annotating data, which the classifier is not able to predict correctly. They show that the classification performance is within 2 % of the performance which is obtained by networks that are trained with laboriously annotated pixel-wise data.

Another approach to get enough data for the training is to create simulated data. The amount of data can help to increase performance and improve the generalization capabilities of the classifier. Di Cicco *et al.* [26] try to reduce the human effort for labeling by constructing synthetic datasets using a physical model of a sugar beet leaf. Their results indicate that such artificially generated datasets can support traditional approaches by providing additional training data. Potena *et al.* [117] reduce the required amount for labeling by an unsupervised dataset summarization procedure before the actual labeling process takes place. The key idea is to select a subset of a fixed size, which gives the most informative description of the whole dataset. Dyrmann [28] presents in his Ph.D. thesis an approach based on fully convolution neural networks that perform a pixel-wise classification into the classes crop, weed, and soil, under natural lighting conditions. For the weeds, this work considers the 17 most common species in Danish fields. About 4,500 pictures are annotated by hand for the training of the networks. The annotated data is then used to artificially create additional images, in which the annotated plants are first cut out of the RGB data and then randomly reinserted into images with only soil pixels. Based on this data augmentation technique, the used convolutional neural network architecture VGG16 [136] archives and overall classification accuracy of around 87 %.

Hall *et al.* [44] argue that one limitation for the deployment of classifiers for weed detection is that they are typically trained on a defined set of known weed types. In practice, however, other weed species may appear in the field. These examples can most likely not be handled by the classifier. Thus, the authors propose a clustering approach to weed scouting, which can be utilized in any field without the need for prior species knowledge. The key idea is to detect patches of plants in the field, cluster them regarding their similarity, and finally assign a semantic label to the detected clusters. First, they detect the vegetation using a multivariate Gaussian classifier based on color features. Then, they extract features using convolutional neural networks for each patch and pass them to the cluster analysis. Hall *et al.* [45] present a system for weed classification for mobile robots. They minimize the labeling effort by using an unsupervised weed scouting process. They first employ the clustering and afterward only label a small number of candidates to label the whole acquired data. Based on the labeled dataset, they train a multi-class linear support vector machine and perform autonomous precision weeding using the trained classifier. With this approach, they can perform selective and species-specific weed control. However, this approach considers that the robot visits the field two times. Once for scouting and once for weed control.

Other researchers take advantage of prior information about the process or the field environment to generate training data semi-automatically. Wendel and Underwood [154] address this by proposing a method for training data generation in order to feed a classifier for crop and weed detection with it. They use a multispectral line scanner mounted on a field robot and perform a vegetation segmentation followed by a crop-row detection. Subsequently, they assign the label crop for the pixels corresponding to the crop-row and the remaining ones as a weed. Rainville *et al.* [123] propose a vision-based method to learn a probability distribution of morphological features based on a previously computed crop-row. Similar to the work of Wendel and Underwood [154], they also exploit crop-row structure as prior to labeling. Bah *et al.* [6] propose a semi-supervised approach for UAV-Based crop and weed classification. They exploit that row crops grow in line. They first classify the vegetation using a threshold-based approach based on the excess green index. Second, they identify the crop rows in the images through performing skeletonization on the binary vegetation mask. Next, they perform a superpixel segmentation in the RGB image and assign vegetation super-pixels that lie on the estimated crop row as crop and vegetation super-pixels that lie between the rows as a weed. Next, they train a fully convolutional neural network based on the Resnet-34 [52] architecture using the labeled super-pixels. Then they deploy the trained network on extracted super-pixels of other images from the dataset and report classification accuracies of around 94 %. They furthermore show that by this semi-supervised labeling procedure, they obtain only a slightly lower performance compared to a run of the classifier that has bee trained by manually labeled super-pixels.

Within our RF-GC approach, we exploit the plant arrangement to re-learn a random forest with minimal labeling effort in a semi-supervised way. We target an in-field labeling effort of approximately one minute for a human operator and do not consider any pre-trained classifier. We achieve this one-minute labeling effort by placing printed markers next to a set of crop plants at the beginning of the row. Based on this information, we can initialize our classification system and can directly obtain predictions. The geometric and visual classifiers complement each other and adapt to changing field conditions in an online manner through retraining parallel to the deployment.

Chapter 8

Conclusion

In this thesis, we proposed and developed novel plant classification systems for the deployment of autonomous agricultural robots such as unmanned ground robots but also aerial vehicles. Our classification systems provide the basis for selective and plant-specific treatments in the field, such as selective spraying or precise mechanical intervention. Furthermore, they can process aerial image data to estimate the spatial distribution of crop plants, weeds, or even different weeds species, and can robustly count plants, even under harsh conditions.

As our first two contributions, we developed two different vision-based plant classification systems. The first classification system is based on handcrafted features using random forests and the second plant classification system is based on a lightweight, fully convolutional neural networks. Both vision-based classification systems identify plants using RGB and near-infrared images or solely RGB images and can deal with small plants and weeds. Also, in situations with substantial inter-class overlap, our classifiers provide a stable performance allowing the robots to intervene with high-level precision. Our random forest-based classifier achieves this by combining local features for keypoints with object- or segment-based features within a cascaded approach. Our fully convolutional neural network approach is based on a self-designed, lightweight encoder-decoder architecture that learns features and performs the classification in an end-to-end manner. We demonstrated that both the random forest-based and the fully convolutional neural network-based classifies can handle multi-class problems, e.g., to predict different weeds species, and can be deployed on real agricultural robots, as they provide the classification results fast enough for online on-field operations.

As our third contribution, we adopted the random forest-based and fully convolutional neural network-based plant classification systems to work with aerial image data. Here, we enhanced the random forest with additional handcrafted features exploiting the crop row structure and spatial relationships between plants and weed in the field. In contrast, we adapt the fully convolutional neural network-based approach by enlarging its receptive field. Through this, we enable the network to learn the spatial patterns in the data implicitly instead of modeling them by hand explicitly. We showed in our experiments that under similar field conditions, i.e., when the classifiers have access to training data from the same field environment, the fully convolutional neural networks perform substantially better than the random forest. Under changing field conditions, however, the explicit modeling of the crop row structure is a key supporter for achieving a robust performance. This outcome suggests also combine such information with a convolutional network, e.g., within a postprocessing step.

As our fourth contribution, we presented a novel end-to-end trainable fully convolutional neural network for joint pixel-wise plant classification and plant stem detection. We designed a network architecture with a shared encoder for the feature extraction and two task-specific decoders for the pixel-wise classification of plant stems. The system enables the robot for high precision and plant-specific treatments in the field. First, the system jointly estimates the pixel-wise segmentation into the classes crop, dicotyl weed, grass weed, and soil. The information about the class and spatial extent of the plants and weeds can be used for the guidance of the selective sprayers. Second, the system estimates the precise locations of plant and dicotyl weed stems. This information can be used to guide precise mechanical tools or even lasers. We showed in our experiments that our architectural design choice to use two task-specific decoders outperforms network architectures suited for only one task, i.e., plant classification or stem detection.

In this thesis, we furthermore aim at bridging the performance gap in visual crop plant and weed detection if the distribution of the features at training time differs from the one observed during operation. This situation commonly arises when a robot equipped with a pre-trained classifier is supposed to perform the classification in new or changing field environments. To address this challenge, we proposed two novel approaches that exploit a large number of crop plants that are sown in rows and share a similar lattice distance along the crop rows. As our fifth contribution, we proposed a semi-supervised online learning approach for the random forest-based classification system. We developed a probabilistic model representing the arrangement of the plants encoded through coordinate differences between plants. We employ this model as a purely geometric classifier and combine it with the visual random forest classifier in a semi-supervised way. Furthermore, our semi-supervised approach has online learning capabilities and thus can adapt itself to new and unseen data. In our experiments, we demonstrate that this approach can perform on the same level with state-of-the-art plant classifiers by only requiring a labeling effort of around one minute for a new field environment.

Our sixth contribution is a novel architectural extension to fully convolutional neural networks that allows the network to process sequences of images using 3D convolutions. We call this extension the sequential module. It exploits successively acquired images to learn features describing the local arrangement of the plants implicitly. We show in our experiments that this technique leads to a better classification and generalization performance, even if the visual appearance or the growth stage of the plants change between training and test time and outperforms classification models that operate on single images.

Our comprehensive experimental evaluation and extensive comparisons represent our seventh contribution of this thesis. Here, we demonstrate that we could substantially improve the generalization performance of plant classification through our approaches that exploit the spatial arrangement of the plants in the field. We showed that the random forest-based classifiers rely more on the additional near-infrared information for accurate vegetation classification. Also, in terms of plant classification, the fully convolutional neural networks generally perform better. We have observed that they can handle a greater diversity of data appropriately. It turns out that neural networks can learn better features than the handcrafted ones in the random forest. However, we also observe that even though the exploitation of the field geometry, the classifiers provide not always reliable results under changing field conditions. For the evaluation of the proposed approaches, we collected an extensive database consisting of approximately 26,500 labeled images. We acquired the data in different fields located in Germany, Switzerland, and Italy and evaluate our approaches in the context of their performance, generalization capabilities, labeling effort, use of additional NIR information, and architectural design choices.

8.1 Future Work

We have gained a deep insight into scientific and practical challenges for vision-based plant classification systems for mobile robots during the compilation of this thesis and the re-running of all experiments with all approaches under the same conditions. The partially remaining challenge for the practical deployment of autonomous systems is the generalization capabilities of classifiers to new and changing field conditions, see Figure 8.1. Moreover, the scalable use of such classification systems can only be achieved if the annotation effort for the adaptation of the model is wholly omitted or only very small.

We showed that vision-based classification systems for plant classification could obtain high classification performances in the order of 90+% in terms of classification accuracy when the training and test data where acquired under similar environmental conditions with the same camera setup. However, the classification performance suffers under substantial changes in the plant appearance and soil conditions between training and testing phase as the classifier, which is trained on labeled data coming from previously seen fields *source domain*, has never learned to handle the different distribution of the data coming from new field environments *target domain.* Here the performance can drop to unsuitable results for the desired intervention. Although we have shown that the relative spatial distribution of plants is an essential feature for addressing this challenge, classifiers have not always been able to deliver consistently high performance in all situations.

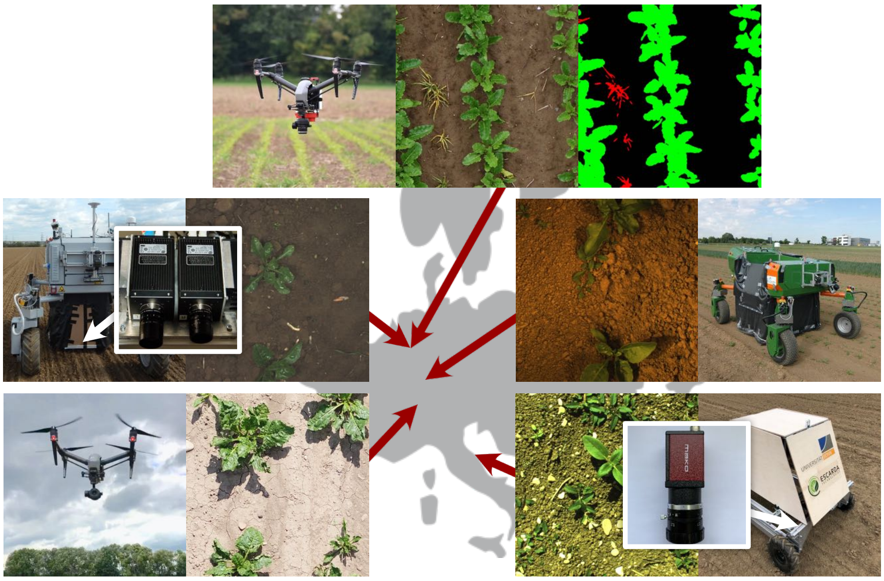


Figure 8.1: Data acquisition under varying conditions, in different fields, and with different cameras and robots leads to highly distinctive image domains that challenge pre-trained plant classification systems to generalize well. We propose an effective approach for adapting existing systems to new environments, different crops, and other different field conditions.

8.2 Outlook: Unsupervised Domain Adaptation for Transferring Plant Classification Systems to New Field Environments, Crops, and Robots

In our recently submitted paper, which is currently under review at the Conference on Robots and Systems (IROS, 2020), we make a first step towards the challenge of unsupervised domain adaptation for transferring plant classification systems to new field environments, crops, and robots. We propose an effective approach to unsupervised domain adaptation for plant segmentation systems in agriculture and thus to adapt existing systems to new environments, different value crops, and other farm robots. We aim to bridge the performance gap in visual crop and weed classification by transferring the visual classifier to the targeted domain without the need for an additional labeling effort. We target unsupervised domain adaptation towards an approach that enables us to train a fully convolutional neural network with suitable performance on the target domain while exploiting labels only from the source domain. This work was done in tight collaboration with Dario Gogoll.

The key idea of our proposed approach is to transfer both images and corresponding labels from a training dataset, i.e., the source domain, into the style of the conditions of the new or changed conditions in the targeted field, i.e., the target domain. We simultaneously train a fully convolutional neural network during this domain transfer using the translated images in the style of the target domain alongside copied labels

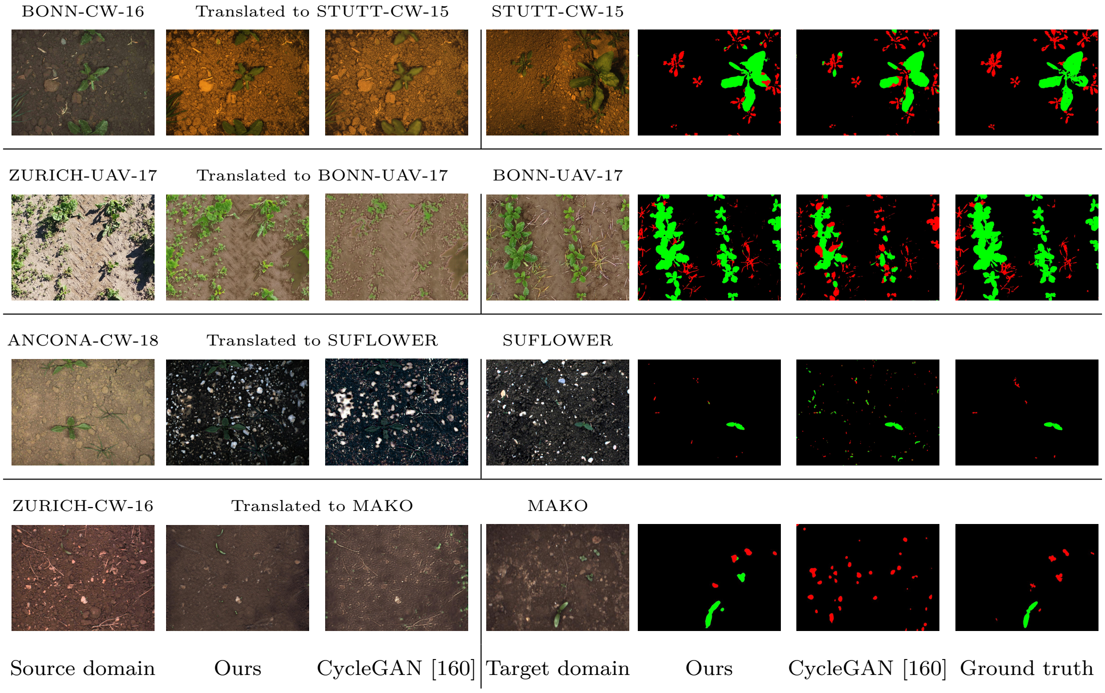


Figure 8.2: Qualitative results of our domain adaptation approach. Our approach exploiting semantics provides a better translation into the target domain compared to CycleGANs. Our approach preserves fine structures and properly transfers the semantic information in a pixel-wise manner. The CycleGAN approach suffers from missing semantic information. It wrongly translates pixels that belong to small vegetation objects or fine structures.

from the source domain in a supervised manner. Our proposed system is based on CycleGANs [160] and enforces a semantically consistent domain transfer by constraining the images to be classified in the same way before and after translation. As a result, our approach generates labeled images of the target domain that enables us to retrain existing segmentation systems.

We design a set of experiments to show that our approach can translate images from one domain into the other and can train a fully convolutional neural network that achieves a high classification performance in the target domain, without the need for labeled data from the target domain. We evaluate our approach on eight different real-world datasets consisting of 6,221 images.

Our evaluation shows that our unsupervised domain adaption approach provides a solid performance for the semantic segmentation of crop, weed, and soil in the target domain, while not requiring extra labels from the target domain for the adaption of the classifier. Furthermore, it outperforms CycleGANs and other baselines on the target domain for all tested datasets. Finally, it allows to perform domain adaptation between different field environment, different crops, and different robots and camera setups. Our approach achieves substantial performance gains of around 20 %-50 % for the pixel-wise classification accuracy in the target domain compared to the classification without domain adaption. Also, it outperforms the original CycleGAN approach. Figure 8.2 illustrates the qualitative results obtained by our proposed domain adaptation approach. The results reveal excellent plant classification results in the target domain. Our approach reliably generates images in the style of the target domain while keeping the source domain images semantics. This, in turn, allows us to train a fully convolutional neural network for the target domain using the translated images along with the original labels from the source domain.

We believe that such domain adaptation approaches are essential to achieve a scalable deployment of agricultural robots in the real world. We think that unsupervised domain adaption is a field that should be further explored by researches in the future.

Bibliography

[1] JAI AD-130-GE camera manual. `https://www.jai.com/downloads/datasheet-ad-130ge`. Accessed: 2019-12-01.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv preprint*, 1502.03167, 2016.

[3] M.J. Aitkenhead, I.A. Dalgetty, C.E. Mullins, A.J.S. McDonald, and N.J.C. Strachan. Weed and crop discrimination using image analysis and artificial intelligence methods. *Computers and Electronics in Agriculture*, 39(3):157–171, 2003.

[4] D. Albani, D. Nardi, and V. Trianni. Field coverage and weed mapping by uav swarms. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4319–4325, 2017.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017.

[6] M.D. Bah, A. Hafiane, and R. Canals. Deep Learning with unsupervised data labeling for weeds detection on UAV images. *arXiv preprint*, 1805.12395v1, 2018.

[7] A. Bechar and C. Vigneault. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111, 2016.

[8] A. Bechar and C. Vigneault. Agricultural robots for field operations. part 2: Operations and systems. *Biosystems Engineering*, 153:110 – 128, 2017.

[9] J. Behley, V. Steinhage, and A.B. Cremers. Laser-based Segment Classification Using a Mixture of Bag-of-Words. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[10] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. *arXiv preprint*, 1206.5533v2, 2012.

[11] I. Bogoslavskyi, O. Vysotska, J. Serafin, G. Grisetti, and C. Stachniss. Efficient Traversability Analysis for Mobile Robots using the Kinect Sensor. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, Barcelona, Spain, 2013.

[12] T. Borregaard, H. Nielsen, L. Norgaard, and H. Have. Crop-weed discrimination by line imaging spectroscopy. *Journal of Agricultural Engineering Research*, 72(4):389–400, 2000.

[13] B.E Boser, I.M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the workshop on computational learning theory (COLT)*, 1992.

[14] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak. Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics (JFR)*, 37(1):7–19, 2020.

[15] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[16] D. Bullock, A. Mangeni, T. Wiesner-Hanks, C. DeChant, E.L. Stewart, N. Kaczmar, J.M. Kolkman, R.J. Nelson, M.A. Gore, and H. Lipson. Automated Weed Detection in Aerial Imagery with Context. *arXiv preprint*, 1910.00652v3, 2019.

[17] T.F. Burks, S.A. Shearer, R.S. Gates, and K.D. Donohue. Backpropagation neural network design and evaluation for classifying weed species using color image texture. *Transactions of the American Society of Agricultural Engineers*, 43(4):1029–1037, 2000.

[18] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, and D. Coquin. A model-based approach for compound leaves understanding and identification. In *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*, pages 1471–1475, 2013.

[19] N. Chebrolu, P. Lottes, T. Laebe, and C. Stachniss. Robot Localization Based on Aerial Images for Precision Agriculture Tasks in Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[20] N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *Intl. Journal of Robotics Research (IJRR)*, 36(10):1045–1052, 2017.

[21] Ł. Chechliński, B. Siemiątkowska, and M. Majewski. A system for weeds and crops identification—reaching over 10 fps on raspberry pi with the usage of mobilenets, densenet and custom modifications. In *IEEE Sensors Journal*, volume 19, 2019.

[22] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv preprint*, 1802.02611v3, 2018.

[23] F. Chollet. *Deep Learning with Python.* Manning, 2017.

[24] S. Cook. *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs.* Morgan Kaufmann Publishers Inc., 1st edition, 2013.

[25] A. Montes de Oca, L. Arreola, A. Flores, J. Sanchez, and G. R. Flores. Low-cost multispectral imaging system for crop monitoring. *Proc. of the Intl. Conf. on Unmanned Aircraft Systems (ICUAS)*, pages 443–451, 2018.

[26] M. Di Cicco, C. Potena, G. Grisetti, and A. Pretto. Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection. *arXiv preprint*, 1612.03019v3, 2016.

[27] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint*, 1603.07285v2, 2016.

[28] M. Dyrmann. *Automatic Detection and Classification of Weed Seedlings under Natural Light Conditions.* PhD thesis, University of Southern Denmark, Department of Computer Science, 2017.

[29] E. Elhariri, N. El-Bendary, and A. E. Hassanien. Plant classification system based on leaf features. In *Proc. of the Intl. Conf. on Computer Engineering Systems (ICCES)*, pages 271–276, 2014.

[30] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intl. Journal of Computer Vision (IJCV)*, 70, October 2006.

[31] F. Feyaerts and L. van Gool. Multi-spectral vision system for weed detection. *Pattern Recognition Letters*, 22:667–674, 2001.

[32] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[33] F.V. Fleckenstein, C. Dornhege, and W. Burgard. Efficient Path Planning for Mobile Robots with Adjustable Wheel Positions. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[34] Y. Freund and R.E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. volume 55, pages 119–139, 1997.

[35] J. Fuentes-Pacheco, J. Torres-Olivares, E. Roman-Rangel, S. Cervantes P., Juarez-Lopezo, J. Hermosillo-Valadez, and J. M. Rendón-Mancha. Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sensing*, 11(10), 2019.

[36] J. Gai, L. Tang, , and B. Steward. Plant Localization and Discrimination using 2D+3D Computer Vision for Robotic Intra-row Weed Control. In *Agricultural and Biosystems Engineering Conference Proceedings and Presentations*, 2016.

[37] J. Geipel, J. Link, and W. Claupein. Combined spectral and spatial modeling of corn yield based on aerial images and crop surface models acquired with an unmanned aircraft system. *Remote Sensing*, 6(11):10335, 2014.

[38] S. Ghosal, B. Zheng, S. C. Chapman, A. B. Potgieter, D. R. Jordan, X. Wang, A. K. Singh, A. Singh, M. Hirafuji, S. Ninomiya, B. Ganapathysubramanian, S. Sarkar, and W.Guo. A weakly supervised deep learning framework for sorghum head detection and counting. *Plant Phenomics*, 9:1544, 2019.

[39] K.U.L.T. Kress Umweltschonende Landtechnik GmbH. Unkrautflieger firma kress. `http://www.hortipendium.de/Datei:Unkrautflieger_Kress.jpg` under license CC BY-NC-SA 3.0 DE. Accessed: 2020-04-06.

[40] H.C.J. Godfray, J.R. Beddington, R.R. Crute, L. Haddad, D. Lawrence, J.F. Muir, J.N. Pretty, S. Robinson, S.M. Thomas, and C. Toulmin. Food security: the challenge of feeding 9 billion people. *Science*, 327 5967:812–8, 2010.

[41] J. M. Guerrero, G. Pajares, M. Montalvo, J. Romeo, and M. Guijarro. Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications*, 39(12):11149 – 11155, 2012.

[42] W. Guo, U.K. Rage, and S. Ninomiya. Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model. *Computers and Electronics in Agriculture*, 96:58–66, 2013.

[43] W. Guo, B. Zheng, A. B. Potgieter, J. Diot K. Watanabe, K. Noshita, D. R. Jordan, X. Wang, J. Watson, S. Ninomiya, and S. C. Chapman. Aerial imagery analysis – quantifying appearance and number of sorghum heads for applications in breeding and agronomy. *Frontiers in Plant Science*, 9:1544, 2018.

[44] D. Hall, F. Dayoub, J. Kulk, and C.S. McCool. Towards Unsupervised Weed Scouting for Agricultural Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[45] D. Hall, F. Dayoub, T. Perez, and C.S. McCool. A Transplantable System for Weed Classification by Agricultural Robotics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[46] D. Hall, C.S. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft. Evaluation of features for leaf classification in challenging conditions. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 797–804, Jan 2015.

[47] E. Hamuda, M. Glavin, and E. Jones. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125:184–199, 2016.

[48] E. Hamuda, M. Glavin, and E. Jones. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125:184–199, 2016.

[49] S. Haug, P. Biber, A. Michaels, and J. Ostermann. Plant stem detection and position estimation using machine vision. In *Workshop Proc. of Conf. on Intelligent Autonomous Systems (IAS)*, pages 483–490, 2014.

[50] S. Haug, A. Michaels, P. Biber, and J. Ostermann. Plant classification system for crop / weed discrimination without segmentation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 1142–1149, 2014.

[51] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2015.

[52] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] J. Hemming and T. Rath. Computer-vision-based weed identification under field conditions using controlled lighting. *Journal of Agricultural Engineering Research*, 78(3):233 – 243, 2001.

[54] L. Horrigan, R. S. Lawrence, and P. Walker. How sustainable agriculture can address the environmental and human health harms of industrial agriculture. *Environ Health Perspect*, 110:445–56, 2002.

[55] G. Huang, Z. Liu, L.v.d. Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[56] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*, 1502.03167, 2015.

[57] G. R. F. Jose, D. Wulfsohn, and J. Rasmussen. Sugar beet (beta vulgaris l.) and thistle (cirsium arvensis l.) discrimination based on field spectral data. *Biosystems Engineering*, 139:1–15, 2015.

[58] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *arXiv preprint*, 1611.09326v3, 2016.

[59] M. Kampen, S. Lederbauer, J. P. Mund, and M. Immitzer. Uav-based multispectral data for tree species classification and tree vitality analysis. In Publikationen der DGPF, editor, *Dreiländertagung der DGPF der OVG und der SGPF*, volume 28, pages 623–639, 2019.

[60] A. Karpathy. *Connecting images and natural language.* PhD thesis, University of Stanford, Stanford Vision Lab, 2016.

[61] R. Khanna, M. Möller, J. Pfeifer, F. Liebisch, A. Walter, and R. Siegwart. Beyond point clouds - 3d mapping and field parameter measurements using uavs. In *Proc. of the IEEE Conf. on Emerging Technologies Factory Automation (ETFA)*, pages 1–4, 2015.

[62] S. Kiani and A. Jafari. Crop detection and positioning in the field using discriminant analysis and neural networks based on shape features. *Journal of Agricultural Science and Technology*, 14:755–765, 2012.

[63] D.P. Kingma and J.Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 1412.6980, 2014.

[64] N. Kirchgessner, F. Liebisch, K. Yu, J. Pfeifer, M. Friedli, A. Hund, and A. Walter. The eth field phenotyping platform fip: A cable-suspended multi-sensor system. *Functional Plant Biology*, 44:154–168, 2017.

[65] F. Kraemer, A. Schaefer, A. Eitel, J. Vertens, and W. Burgard. From Plants to Landmarks: Time-invariant Plant Localization that uses Deep Pose Regression in Agricultural Fields. In *IROS Workshop on Agri-Food Robotics*, 2017.

[66] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2012.

[67] N. Kumar, P.N. Belhumeur, A. Biswas, D.W. Jacobs, W.J. Kress, I. Lopez, and J.V.B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2012.

[68] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. A Navigation System for Robots Operating in Crowded Urban Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2013.

[69] F. Langer, L. Mandtler, A. Milioto, E. Palazzolo, and C. Stachniss. Geometrical Stem Detection from Image Data for Precision Agriculture. *arXiv preprint*, 1812.05415v1, 2018.

[70] M. V. Latte, B. S. Anami, and V. B. Kuligod. A combined color and texture features based methodology for recognition of crop field image. *Intl. Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(2):287–302, 2015.

[71] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino. Deep-plant: Plant identification with convolutional neural networks. In *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*, pages 452–456, 2015.

[72] C. Lehnert, A. English, C.S. McCool, A.M.W. Tow, and T. Perez. Autonomous Sweet Pepper Harvesting for Protected Cropping Systems. *IEEE Robotics and Automation Letters (RA-L)*, 2(2):872–879, 2017.

[73] C. Lehnert, I. Sa, C.S. McCool, B. Upcroft, and T. Perez. Sweet Pepper Pose Detection and Grasping for Automated Crop Harvesting. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[74] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Intl. Journal of Computer Vision (IJCV)*, 2008.

[75] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[76] X. Li, H. Chen, X. Qi, Q. Dou, C. Fu, and P.A. Heng. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes. *arXiv preprint*, 1709.07330v3, 2017.

[77] F. Liebisch, J. Pfeifer, R. Khanna, P. Lottes, C. Stachniss, T. Falck, S. Sander, R. Siegwart, A. Walter, and E. Galceran. Flourish – A robotic approach for automation in crop management. In *In Proc. of the Workshop für Computer-Bildanalyse und unbemannte autonom fliegende Systeme in der Landwirtschaft*, 2016.

[78] F. Liebisch, M. Popovic, J. Pfeifer, R. Khanna, P. Lottes, C. Stachniss, A. Pretto, S. In Kyu, J. Nieto, R. Siegwart, and A. Walter. Automatic uav-based field inspection campaigns for weeding in row crops. In *Proceedings of the 10th EARSeL SIG Imaging Spectroscopy Workshop*, 2017.

[79] T.Y. Lin, P. Goyal, R.B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint*, 1708.02002, 2017.

[80] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[81] P. Lottes. Bildbasierte klassifikation von zuckerrüben und unkräutern für mobile roboter. Master's thesis, University of Bonn, Department of Photogrammetry, 2015. In German.

[82] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[83] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Robust joint crop-weed classification and stem detection using image sequences. *Journal of Field Robotics (JFR)*, 37(1):20–34, 2020.

[84] P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 3:3097–3104, 2018.

[85] P. Lottes, N. Chebrolu, F. Liebisch, and C. Stachniss. UAV-based field monitoring for precision farming. In *25. Workshop Computer-Bildanalyse in der Landwirtschaft*, 2019.

[86] P. Lottes, M. Höferlin, S. Sander, M. Müter, P. Schulze-Lammers, and C. Stachniss. An Effective Classification System for Separating Sugar Beets and Weeds for Precision Farming Applications. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[87] P. Lottes, M. Höferlin, S. Sander, and C. Stachniss. Effective Vision-based Classification for Separating Sugar Beets and Weeds for Precision Farming. *Journal of Field Robotics (JFR)*, 34:1160–1178, 2017.

[88] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss. UAV-Based Crop and Weed Classification for Smart Farming. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[89] P. Lottes and C. Stachniss. Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[90] C. McCool, J. Beattie, M. Milford, J. D. Bakker, J. L. Moore, and J. Firn. Automating analysis of vegetation with computer vision: Cover estimates and classification. *Ecology and Evolution*, 8(12):6005–6015, 2018.

[91] C.S. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, R. Russell, T. Perez, and O. Bawden. Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters (RA-L)*, 3(2):1184–1190, 2018.

[92] C.S. McCool, T. Perez, and B. Upcroft. Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics. *IEEE Robotics and Automation Letters (RA-L)*, 2(3):1344–1351, 2017.

[93] C.S. McCool, I. Sa, F. Dayoub, C. Lehnert, T. Perez, and B. Upcroft. Visual Detection of Occluded Crop: For Automated Harvesting. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[94] H. S. Midtiby and J. Rasmussen. Automatic location of crop rows in uav images. In *NJF Seminar 477. Future arable farming and agricultural engineering*, pages 22–25, 2014.

[95] H.S. Midtiby, T.M. Giselsson, and R.N. Joergensen. Estimating the plant stem emerging points (pseps) of sugar beets at early growth stages. *Biosystems Engineering*, 111(1):83 – 90, 2012.

[96] A. Milioto, P. Lottes, and C. Stachniss. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W3, pages 41–48, 2017.

[97] A. Milioto, P. Lottes, and C. Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[98] A. Milioto and C. Stachniss. Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[99] M. Montalvo, G. Pajares, J. M. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. J. Ruz, and J. M. Cruz. Automatic detection of crop rows in maize fields with high weeds pressure. *Expert System Applications*, 39(15):11889–11897, 2012.

[100] A.K. Mortensen, M. Dyrmann, H. Karstoft, R. N. Jörgensen, and R. Gislum. Semantic Segmentation of Mixed Crops using Deep Convolutional Neural Network. In *Proc. of the Intl. Conf. of Agricultural Engineering (CIGR)*, 2016.

[101] M. Müter, P. Schulze Lammers, and L. Damerow. Development of an intra-row weeding system using electric servo drives and machine vision for plant detection. In *Proc. of the Agricultural Engineering Conference*, 2013.

[102] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 31–45, 2016.

[103] A.T. Nieuwenhuizen. *Automated detection and control of volunteer potato plants.* PhD thesis, Wageningen University, 2009.

[104] N.Otsu. A tlreshold selection method from gray-level histograms. *Proc. of the IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*, 9(1):62–66, 1979.

[105] T. Ojala and M. Pietikääinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, (32):477–486, 1999.

[106] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. Rahimi Azghadi, and R. D. White. DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning. *Scientific Reports*, 9(2058), 2019.

[107] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 1988.

[108] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv preprint*, 1606.02147v1, 2016.

[109] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 8024–8035, 2019.

[110] S. M. Pedersen, S. Fountas, H. Have, and B. S. Blackmore. Agricultural robots—system analysis and economic feasibility. *Precision Agriculture*, 7:295–308, 2006.

[111] J. M. Peña, J. Torres-Sánchez, A. I. de Castro, M. Kelly, and F. López-Granados. Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle uav images. *PLoS ONE*, 8, 2013.

[112] J. M. Peña, J. Torres-Sánchez, A. Serrano-Perez, A. I. de Castro, and F. López-Granados. Quantifying efficacy and limits of unmanned aerial vehicle uav technology for weed seedling detection as affected by sensor resolution. *IEEE Sensors Journal*, 15(3), 2015.

[113] M. Perez-Ortiz, J. M. Peña, P. A. Gutierrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados. A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method. *Applied Soft Computing*, 37:533 – 544, 2015.

[114] M. Perez-Ortiz, J. M. Peña, P. A. Gutierrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados. Selecting patterns and features for between- and within- crop-row weed mapping using uav-imagery. *Expert Systems with Applications*, 47:85 – 94, 2016.

[115] J. Pfeifer, R. Khanna, C. Dragos, M. Popovic, E. Galceran, N. Kirchgessner, A. Walter, R. Siegwart, and F. Liebisch. Towards automatic uav data interpretation for precision farming. In *Proc. of the Intl. Conf. of Agricultural Engineering (CIGR)*, 2016.

[116] M. Popovic, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran. Online Informative Path Planning for Active Classification Using UAVs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[117] C. Potena, D. Nardi, and A. Pretto. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *Proc. of Int. Conf. on Intelligent Autonomous Systems (IAS)*, 2016.

[118] M. P. Pound, J. A. Atkinson, D. M. Wells, T. P. Pridmore, and A. P. French. Deep learning for multi-task plant phenotyping. In *Proc. of the Int. Conf. on Computer Vision (ICCV) Workshops*, pages 2055–2063, 2017.

[119] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebisch, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popovic, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer an R. Siegwart, C. Stachniss, A. Walter, V. Winterhalter, X. Wu, and J. Nieto. Building an Aerial-Ground Robotics Systemfor Precision Farming. *IEEE Robotics & Automation Magazine*, 2020.

[120] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebisch, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popović, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer, R. Siegwart, C. Stachniss, A. Walter, W. Winterhalter, X. Wu, and J. Nieto. Building an Aerial-Ground Robotics System for Precision Farming. *arXiv preprint*, 2019.

[121] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A.Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[122] M. A. Rahman and Y. Wang. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In *Int. Symp. on Visual Computing*, 2016.

[123] F.M. De Rainville, A. Durand, F.A. Fortin, K. Tanguy, X. Maldague, B. Panneton, and M.J. Simard. Bayesian classification and unsupervised learning for isolating weeds in row crops. *Pattern Analysis and Applications*, 17(2):401–414, 2014.

[124] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint*, 1804.02767v1, 2018.

[125] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. on Intelligent Transportation Systems (ITS)*, 19(1):263–272, 2018.

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

[127] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering. Monitoring Vegetation Systems in the Great Plains with Erts. *NASA Special Publication*, 351:309, 1974.

[128] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[129] I. Sa, Z. Chen, M. Popovic, R. Khanna, F. Liebisch, J. Nieto, and R. Siegwart. Weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robotics and Automation Letters (RA-L)*, 3(1):588–595, 2018.

[130] I. Sa, C. Lehnert, A. English, C.S. McCool, F. Dayoub, B. Upcroft, and T. Perez. Peduncle Detection of Sweet Pepper for Autonomous Crop Harvesting - Combined Colour and 3D Information. *IEEE Robotics and Automation Letters (RA-L)*, 2(2):765–772, 2017.

[131] I. Sa, M. Popovic, R. Khanna, Z. Chen, P. Lottes, F. Liebisch, J. Nieto, C. Stachniss, and R. Siegwart. WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming. *Remote Sensing*, 10, 2018.

[132] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv preprint*, 1801.04381v3, 2018.

[133] R. Shamshiri, C. Weltzien, I. A. Hameed, I. J. Yule, T. E. Grift, and S. K. Balasundram. Research and development in agricultural robotics: A perspective of digital farming. *Intl. Journal of Agricultural and Biological Engineering*, 11:1–14, 2018.

[134] S.A. Shearer and R.G. Holmes. Plant identification using color co-occurrence matrices. *Transactions of the American Society of Agricultural Engineers*, 33(6):2037–2044, 1990.

[135] R. Sheikh, A. Milioto, P. Lottes, C. Stachniss, M. Bennewitz, and T. Schultz. Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[136] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, 1409.1556, 2014.

[137] D.C. Slaughter, D.K. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1):63 – 78, 2008.

[138] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal on Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[139] C. Stachniss, O. Martínez-Mozos, A. Rottmann, and W. Burgard. Semantic Labeling of Places. In *Proc. of the Intl. Symposium on Robotic Research (ISRR)*, 2005.

[140] W. Strothmann, A. Ruckelshausen, J. Hertzberg, C. Scholz, and F. Langsenkamp. Plant classification with in-field-labeling for crop/weed discrimination using spectral features and 3d surface features from a multi-wavelength laser line profile system. *Computers and Electronics in Agriculture*, 134:79 – 93, 2017.

[141] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[142] N. Teimouri, M. Dyrmann, P. R. Nielsen, S. K. Mathiassen, G. J. Somerville, and R. N. Jørgensen. Weed growth stage estimator using deep convolutional neural networks. *Sensors*, 18(5), 2018.

[143] A. Tellaeche, X.P. Burgos-Artizzu, G. Pajares, and A. Ribeiro. A vision-based method for weeds identification through the bayesian decision theory. *Pattern Recognition*, 41(2):521–530, 2008.

[144] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[145] D. Tilman, C. Balzer, J. Hill, and B. L. Befort. Global food demand and the sustainable intensification of agriculture. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 108, pages 20260–4, 2011.

[146] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler. *Sensor planning for a symbiotic UAV and UGV system for precision agriculture*, pages 5321–5326. 2013.

[147] J. Torres-Sanchez, F. López-Granados, and J.M. Peña. An automatic object-based method for optimal thresholding in uav images: Application for vegetation detection in herbaceous crops. *Computers and Electronics in Agriculture*, 114:43 – 52, 2015.

[148] P. Viola and M. Jones. Robust real-time object detection. *Intl. Journal of Computer Vision (IJCV)*, 2001.

[149] S.G. Vougioukas. Agricultural robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1), 2019.

[150] A. Walter, R. Finger, R. Huber, and N. Buchmann. Opinion: Smart farming is key to developing sustainable agriculture. *Proceedings of the National Academy of Sciences*, 114(24):6148–6150, 2017.

[151] A. Walter, R. Khanna, P. Lottes, C. Stachniss, R. Siegwart, J. Nieto, and F. Liebisch. Flourish - a robotic approach for automation in crop management. In *Proc. of the Intl. Conf. on Precision Agriculture*, 2018.

[152] X.-F. Wang, D. Huang, J. Du, H. Xu, and L. Heutte. Classification of plant leaf images with complicated background. *Applied Mathematics and Computation*, 205:916–926, 2008.

[153] U. Weiss and P. Biber. Plant detection and mapping for agricultural robots using a 3d lidar sensor. *Robotics and Autonomous Systems*, 59(5):265–273, 2011.

[154] A. Wendel and J.P. Underwood. Self-Supervised Weed Detection in Vegetable Crops Using Ground Based Hyperspectral Imaging. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[155] W. Winterhalter, F. V. Fleckenstein, C. Dornhege, and W. Burgard. Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters (RA-L)*, 3:3394–3401, 2018.

[156] X. Wu, S. Aravecchia, P. Lottes, C. Stachniss, and C. Pradalier. Robotic weed control using automated weed and crop classification. *Journal of Field Robotics (JFR)*, 37:322–340, 2020.

[157] K.M. Wurm, H. Kretzschmar, R. Kümmerle, C. Stachniss, and W. Burgard. Identifying Vegetation from Laser Data in Structured Outdoor Environments. *Journal on Robotics and Autonomous Systems (RAS)*, 62(5):675 – 684, 2014.

[158] J. Yu, A. Schumann, Z. Cao, S. M. Sharpe, and N. S. Boyd. Weed detection in perennial ryegrass with deep learning convolutional neural network. *Frontiers in Plant Science*, 10:1422, 2019.

[159] X. Zhao, Y. Yuan, M. Song, Y. Ding, F. Lin, D. Liang, and Dongyan Zhang. Use of unmanned aerial vehicle imagery and deep learning unet to extract rice lodging. In *IEEE Sensors Journal*, volume 19, 2019.

[160] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 2223–2232, 2017.

List of Figures

List of Tables