Fortschritt-Berichte VDI

VDI

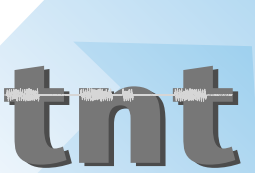
Reihe 10

Informatik/
Kommunikation

Dipl.-Ing. Sebastian Haug,
Stuttgart

Nr. 870

Plant Classification and Position Estimation for Autonomous Field Robots



Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

Plant Classification and Position Estimation for Autonomous Field Robots

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)

genehmigte

Dissertation

von Herrn

Dipl.-Ing. Sebastian Alexander Haug

geboren am 24. Oktober 1986 in Stuttgart

2020

Hauptreferent:	Prof. Dr.-Ing. Jörn Ostermann
Korreferent:	Prof. Dr.-Ing. Holger Blume
Vorsitzender:	Prof. Dr.-Ing. Bodo Rosenhahn

Tag der Promotion: 4. August 2020

Fortschritt-Berichte VDI

Reihe 10

Informatik/
Kommunikation

Dipl.-Ing. Sebastian Haug,
Stuttgart

Nr. 870

Plant Classification and Position Estimation for Autonomous Field Robots

Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

Haug, Sebastian
Plant Classification and Position Estimation for Autonomous Field Robots
Fortschr.-Ber. VDI Reihe 10 Nr. 870. Düsseldorf: VDI Verlag 2020.
164 Seiten, 85 Bilder, 15 Tabellen.
ISBN 978-3-18-387010-3, ISSN 0178-9627,
€ 62,00/VDI-Mitgliederpreis € 55,80.

Keywords: Computer Vision – Machine Learning – Precision Agriculture – Robotics – Plant Classification – Plant Position Estimation – Weed Control

This work presents new approaches to plant classification and plant position estimation to enable field robot based precision agriculture. The developed methods are designed for challenging real world field situations with small crop plants, presence of close-to-crop weed and overlap of plants. The plant classification system is able to distinguish two or more plant classes in field images without the need for error-prone plant or leaf segmentation. The plant position estimation pipeline solves the generic problem of determining the position of both crop and weed plants only from image data. The combination of both methods allows field robots to autonomously determine the type and position of plants in the field to realize precision agriculture tasks such as single plant weed control. Experiments with a field robot prove the applicability of the presented methods for challenging field scenarios encountered for example in organic vegetable farming.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

ISSN 0178-9627

ISBN 978-3-18-387010-3

Vorwort

Die vorliegende Arbeit entstand während meiner Doktorandenzeit in der Forschung und Vorausentwicklung der Robert Bosch GmbH und des Instituts für Informationsverarbeitung an der Gottfried Wilhelm Leibniz Universität Hannover.

Mein besonderer Dank gilt Herrn Professor Dr.-Ing. Jörn Ostermann für die Betreuung meiner Arbeit und die Übernahme des Hauptreferats. Seine konstruktiven und wertvollen Anregungen in unseren Rücksprachen sowie seine Unterstützung trugen maßgeblich zum Erfolg der Arbeit bei.

Herrn Professor Dr.-Ing. Holger Blume möchte ich für die Gespräche zu meiner Arbeit und die Übernahme des Korreferats danken. Herrn Professor Dr.-Ing. Bodo Rosenhahn danke ich für die Übernahme des Vorsitzes.

Insbesondere möchte ich Herrn Dr. rer. nat. Peter Biber für die Betreuung meiner Doktorandenzeit bei Bosch, seine Unterstützung und die fachlichen Diskussionen danken. Herrn Dr. rer. nat. Klaus Marx danke ich für seine Förderung und dass ich in der Abteilung Future Systems Consumer Goods als Doktorand arbeiten konnte.

Allen meinen Kollegen und Kolleginnen bei Bosch gilt mein Dank für die gute Zusammenarbeit und die klasse Zeit. Weiterhin danke ich den Kollegen am Institut für Informationsverarbeitung der Gottfried Wilhelm Leibniz Universität Hannover für den fachlichen Austausch und die interessanten Diskussionen während meiner Besuche und der Klausurtagung des Instituts.

Nicht zuletzt möchte ich allen Mitarbeitern des öffentlich geförderten Projekts Remote-Farming.1 danken: Die gemeinsamen Tage auf dem Feld während der Datenaufnahmen und den zahlreichen Feldtests waren erlebnisreich und legten die Datengrundlage für diese Arbeit und gemeinsame Publikationen. Ebenso danke ich allen Studierenden, die ich während meiner Doktorandenzeit betreuen durfte, für ihre Arbeit und Mitwirkung.

Mein spezieller Dank gilt meiner Familie und meinen Freunden für die Unterstützung und motivierenden Worte in den vergangen Jahren auf dem Weg zum Abschluss meiner Arbeit.

Contents

Symbols and Abbreviations

Symbols

I	Image
h, w	Height (h) and width (w) of an image
u, v	Image pixel coordinates (origin in upper left corner)
p or $p_{u,v}$	Image pixel, u and v determine the coordinates
P	Image patch
w_{size}	Size of a square image patch
w_{stride}	Stride of image patches for example in a sliding window
x, y, z	Right handed coordinate system axis in 3D
σ	Standard Deviation
σ^2	Variance
μ	Mean

Machine Learning

C_i	Class i
$\mathcal{C}$	Set of all classes C_i
f_i	Feature i
$\boldsymbol{f}$	Feature vector
l	Label
g	Ground truth label
$\boldsymbol{s}$	Normalized score vector: $\sum s_i = 1$

Vegetation Segmentation

I_{NDVI}	NDVI image
κ	Scaling factor for NIR intensities
t_{otsu}	Vegetation segmentation threshold
t_{area}	Minimal blob size threshold
t_{nir}	Minimal NIR intensity threshold
$I_{\text{NDVI masked}}$	Vegetation segmented NDVI image

Plant Classification

k_i	Keypoint, describes a pixel location (u_i, v_i)
$\mathcal{K}$	Set of all keypoints k_i
$\mathcal{N}$	Set of neighboring keypoints, 8-connected neighborhood
$\mathcal{N}^+$	Extended neighborhood
λ	Smoothing parameter
L	Labeling, i.e. a specific configuration of labels $[l]$
$\hat{L}$	Smoothed labeling
$\hat{l}$	Smoothed label l
$\hat{L}_{\text{interp}}$	Smoothed labeling interpolated to full image

Plant Position Estimation

s	Stem score (0.0 → no stem, 1.0 → stem)
$\boldsymbol{S}$	Stem score matrix
$\hat{\boldsymbol{S}}$	Smoothed stem score matrix
d_{border}	Size of border around ground truth stems (Chebyshev distance)
$d_{\max}$	Maximum Chebyshev distance of positive samples from stem
d_{step}	Sampling step size for positive patches in window of size $2 \cdot d_{\max}$
$\boldsymbol{g}$	Ground truth stem position at pixel coordinates $(u_{\text{g}}, v_{\text{g}})$
$\boldsymbol{p}$	Detected stem position at pixel coordinates $(u_{\text{p}}, v_{\text{p}})$
γ	Stem detection threshold
k_{smooth}	Smoothing kernel size
$k_{\text{non\_max}}$	Non-maximum suppression kernel size

Mathematical Notation

x	Scalar
$\boldsymbol{x}$	Vector $\boldsymbol{x} = [x_1,\ x_2,\ \ldots,\ x_n]$ with n elements
$\boldsymbol{x}^{\text{T}}$	Column vector: $\boldsymbol{x}^{\text{T}} = [x_1;\ x_2;\ \ldots;\ x_n]$
x_i	i-th element of column vector $\boldsymbol{x}$
$\boldsymbol{X}$	Matrix $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	Inverse of matrix $\boldsymbol{X}$
$\boldsymbol{x}_i$	i-th row of matrix $\boldsymbol{X}$ written as row vector
$x_{i,j}$	Scalar located at i-th row and j-th column of matrix $\boldsymbol{X}$
$\text{argmin}_\phi\, f(x\|\phi)$	Argument ϕ for which the function $f(x)$ in minimal
$\lfloor i \rfloor$	Floor (i.e. next lower integer value) of i
pp	Percentage Point

Abbreviations

ASIC	Application Specific Integrated Circuit
CCD	Charge-Coupled Device
CPU	Central Processing Unit
CRF	Conditional Random Field
CWFID	Crop Weed Field Image Dataset
DB	Database
DVI	Difference Vegetation Index
EGI	Excess Green Index
FA	False Alarm
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
GPU	Graphics Processing Unit
HSI	Hue Saturation Intensity
LDA	Linear Discriminant Analysis
LIDAR	Light Detection And Ranging
MRF	Marcov Random Field
NDI	Normalized Difference Index
NDVI	Normalized Difference Vegetation Index
NIR	Near-Infrared
PCA	Principal Component Analysis
RF	Random Forest
RGB	Red Green Blue
RTK GPS	Real Time Kinematic Global Positioning System
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SSM	Site Specific Management
SVM	Support Vector Machine
UV	Ultra Violet
VI	Vegetation Index
VSLAM	Visual Simultaneous Localization And Mapping

Abstract

Agricultural robots which perceive and understand the field situation enable new, more ecological and sustainable precision agriculture processes. Especially in organic or vegetable crops weed is a major cost source, both through yield loss and when weed control is performed. However, plant classification and position estimation in such high value crops is especially challenging in early growth stage since the crop is typically still small, close-to-crop weed plants of all sizes appear and severe overlap of plants is present.

This thesis presents a new plant classification system and a novel plant position estimation pipeline to enable precision agriculture with field robots. Additionally, a camera system and a vegetation segmentation method are developed. The combined system is finally integrated into a field robot and evaluated in a commercial organic carrot farm.

The novel plant classification system is able to distinguish two or more plant classes in vegetation segmented field images without the need for error-prone plant or leaf segmentation. Feature extraction and supervised classification of overlapping image patches allow the pipeline to handle overlap of plants and irregular shaped leaves. The newly introduced smoothing and interpolation steps compensate the loss of spatial output precision of previously known cell-based methods and ensure a full per-pixel labeled plant classification output image.

The presented plant position estimation pipeline applies a sliding window-based classification approach combined with non-maximum suppression to determine plant stem positions. The pipeline solves the generic problem of determining the position of both crop and weed plants in images only. No additional data such as GPS or crop row information is required. The proposed solution has the advantage that it is applicable to real world field images and not only controlled lab or greenhouse setups. The output plant position estimates are not only suitable for weed control, but also for crop counting and other precision agriculture tasks.

In experiments with the custom built field robot the applicability of the presented methods is proven. In combination with a weed regulation module and the Bonirob field robot, single plant organic weed control in commercial carrot farms is demonstrated.

Keywords: Computer Vision, Machine Learning, Agriculture, Robotics, Plant Classification, Plant Position Estimation, Weed Control

Kurzfassung

Agrarroboter, welche die Feldsituation verstehen, ermöglichen umweltfreundlichere und nachhaltigere Präzisionslandwirtschaft. Insbesondere beim Anbau von Bio- oder hochwertigen Nahrungsmitteln ist Unkraut ein Hauptkostentreiber, sowohl durch Ernteeinbußen als auch durch die Kosten der Unkrautregulierung. Jedoch gerade bei hochwertigen Nahrungsmitteln ist eine automatische Klassifikation und Positionsschätzung in frühem Wachstumsstadium herausfordernd: Nutzpflanzen sind typischerweise noch klein, Unkraut tritt in allen Größen und direkt neben Nutzpflanzen auf. Darüber hinaus kann starke Überlappung von Pflanzen oder Pflanzenteilen vorliegen.

Diese Arbeit stellt ein neues Pflanzenklassifikationssystem und eine neue Methode zur Pflanzenpositionsschätzung vor. Ziel ist Präzisionslandwirtschaft mit Feldrobotern zu ermöglichen. Zusätzlich werden ein Kamerasystem und eine Methode zur Vegetationssegmentierung entwickelt. Abschließend wird das Gesamtsystem in einen Feldroboter integriert und in einer kommerziellen Bio-Karottenfarm evaluiert.

Das entwickelte Pflanzenklassifikationssystem kann zwei oder mehr Pflanzenarten in vegetationssegmentierten Bildern unterscheiden; die fehleranfällige und in verwandten Arbeiten oft benötigte Pflanzen- oder Blattsegmentierung ist hier nicht erforderlich. Merkmalsextraktion und überwachte Klassifikation erfolgt auf überlappenden Bildausschnitten. Dies erlaubt dem System die Verarbeitung von Überlappungen und von Blättern mit unregelmäßiger Form. Die neuen Glättungs- und Interpolationsschritte verhindern den Präzisionsverlust bereits bekannter zellbasierter Methoden und stellen gleichzeitig die Ausgabe eines kompletten Pflanzenklassifikationsbildes sicher.

Die entworfene Methode zur Schätzung von Pflanzenpositionen nutzt ein Klassifikationsverfahren auf Bildausschnitten mit Nichtmaxima-Unterdrückung. Das System löst das generische Problem, die Position von Unkraut und Nutzpflanze zu bestimmen. Es werden außer dem Bild keine zusätzlichen Informationen wie GPS oder Ort der Pflanzenreihe benötigt. Die entwickelte Lösung hat den Vorteil, dass sie auf Feldbilder anwendbar ist und nicht nur in kontrollierten Labor- oder Gewächshausumgebungen funktioniert. Die Positionsausgabe ist über Unkrautregulierung hinaus auch für das Zählen von Pflanzen und andere Methoden der Präzisionslandwirtschaft anwendbar.

Versuche mit dem entsprechend gebauten Feldroboter Bonirob zeigen die Anwendbarkeit des Systems. In Kombination mit einem Unkrautregulierungswerkzeug wird die Regulierung einzelner Unkräuter in einer Bio-Karottenfarm erfolgreich demonstriert.

Stichworte: Bildverarbeitung, Maschinelles Lernen, Landwirtschaft, Robotik, Pflanzenklassifikation, Pflanzenpositionsschätzung, Unkrautregulierung

1 Introduction

Agriculture throughout the world faces major challenges: The need for food keeps growing, arable land is limited and excessive use of chemicals or fertilizer severely impacts the environment. Additionally, new technologies like biofuel or construction with renewable resources put additional pressure on agriculture and forestry around the world. Recent developments in automation, computer science and robotics are promising technologies to cope with these challenges and mitigate negative effects on the environment and climate.

Since more than 30 years precision agriculture techniques for improved farm management have been researched [1]: In precision agriculture variation in the farming process is actively managed with the goal of optimizing the output of the process. For example in site specific management (SSM) fields are not treated homogeneously, rather the process is adjusted to the specific need at the currently treated location in the field [2]. Such site specific management techniques were developed for fertilization, watering, sowing and selective weed control.

Field robots combine advances in information technology, robotics and agriculture to enable high precision farming [3, 4]. For example drones or ground-based robots can generate weed maps [5] and fertilization or weed control processes can be adjusted with high precision for single plants.

Especially in organic vegetable farming of crops such as carrots or onions, severe weed infestation can occur because no chemical herbicides are permitted. Moreover, weed control is required in the early crop growth stages to avoid substantial yield loss [6]. The state of the art in so-called close-to-crop weed control for organic vegetable farming is still manual weed removal by field workers. The weed control task is very tedious, costly and time consuming [7].

The incentive for automating weed control with field robots is manifold and goes beyond organic farming or cost optimization: Excessive use of chemicals for weed treatment threatens the environment and an increase in weeds with resistance against known herbicides [8] questions current farming practices. Additionally, stricter environmental protection legislation enforces farmers to use smaller quantities of chemical herbicides [9] or fertilizers and increases the trend towards novel solutions for the weed problem.

Precision weed control projects share with other activities for intelligent farm management and precision agriculture a requirement for detailed information about the farming process [10]. Such information includes climate, current weather and soil conditions (for example

moisture content and fertility), seed and plant properties (for example species, size, health), data about the agricultural machine and treatment process and many others. An increase in available data enables more automatic, environmentally friendly, cost-effective and organic production of crops and vegetables [11].

Of special interest for precision agriculture is data about individual plants including the type of each plant and its position in the field. A variety of data acquisition methods can be used to collect such information [12]: From manual gathering of measurements, sensors placed at fixed locations in the field, airborne or satellite-based sensing or fully automated data retrieval onboard of ground-based tractors or robots, many sensing concepts have been applied. With airborne or satellite-based sensing large scale imagery can be acquired easily at the cost of reduced resolution [13], whereas ground-based tractor or robot mounted image sensors can deliver high resolution data about single plants or small field patches [14, 15].

Therefore, automated ground-based vision sensors are an especially promising technology to capture these important high resolution measurements in the field and to enable smart precision agriculture projects.

1.1 Scope

This thesis considers automatic acquisition and processing of field images with machine vision to extract information about plants with the goal of enabling novel smart farming applications like weed control in organic farming.

A first goal is to develop and deploy a suitable camera system which needs to be adapted to both the agricultural task as well as how it is deployed to the field.

Second, the objective for the computer vision system is to remove all non-vegetation pixels (for example soil in the background) and then to extract detailed plant related information: On the one hand, this comprises the plant classification task, where plants are classified into individual species or categories like crop and weed. On the other hand, this includes the task of determining the position of a plant in the field.

Approaching these tasks, it has to be considered that in so far unsolved precision agriculture tasks like weed control in organic crops, plants are typically small when weed treatment has to be applied: The plants are still in early growth stage (0 cm to 5 cm in diameter) and parts of plants such as the stem can be as small as 2 mm. Thus, the images must be captured with high resolution sensors. An additional challenge arises because both crop and weed plants have the same size since typically before the crop germinates the complete field is weeded completely for example mechanically or with flame weeders.

Third, the acquisition and extraction of information should happen in an automated manner: Automatic machine vision algorithms are applied to extract the plant properties without human supervision during application in the field.

Fourth, to realize smart farming applications the camera and machine vision system must be compatible to be deployed with an autonomous field robot. When the sensing and computer vision system is combined with for example a weed regulation module, automatic organic weed control can be realized.

1.2 Related Work

The application of computer vision and machine learning to agricultural problems is an active research field: In the following the state of the art for plant classification, plant detection and position estimation is reviewed. A detailed analysis and in depth discussion of related work follows in the individual chapters of the thesis.

Plant Classification The classification of plants or leaves with camera sensors and machine vision has been studied on different levels.

Leaf classification has been researched in constrained scenarios where an image of a single flattened leaf is classified [16, 17, 18, 19, 20]. Some groups developed smartphone applications for leaf classification [21, 22]. Few work has focused on in the field leaf recognition [23, 24, 25].

Ground based plant classification with camera sensors can be divided into three major approaches: First, plant segmentation based methods try to initially segment the field image into blobs that represent single plants and then derive a plant classification decision per blob [26, 27, 28, 29, 30]. These approaches have problems when plants grow close together and overlap. Second, methods that detect the crop row and then use this information to locate weeds were developed [31, 32, 33, 5, 34]. Such row-based methods are not well suited for intra-row weed control where also weed plants within the crop row must be treated. Third, cell-based methods avoid segmentation and tessellate the whole image into non overlapping cells. The classification decision is then output per cell [35, 36, 37, 38, 39]. This inherently reduces the output precision because the decision is limited to whether a cell contains weed or not.

Besides these major directions also other methods have been used to discriminate plants: For example remote sensing [40] and hyperspectral sensing [41, 42, 43] or methods based on crop seed mapping with GPS sensors [44, 45, 46] which enable the system to relocate crops in the field at later stages. Weed control is not the only application for such machine vision pipelines. Also defects on vegetables and fruits [47], diseases on flowers [48] can be detected and robotic harvesting of individual crops can be implemented [49]. Additionally, other agricultural metrics like plant height, nitrogen content, etc. can be derived from images using machine vision [50, 51, 52].

The existing work on plant classification lacks the ability to robustly process field images where plants are of different sizes, overlap heavily and crop plants mix with as well as grow close to weed plants. Furthermore, the plant segmentation and cell-based methods — which better cope with these situations — have the disadvantage that no complete plant

classification image is obtained and therefore single plant weed removal or phenotyping applications are not possible.

Plant Detection and Position Estimation The detection of plants and the estimation of their growing position in fields is an important property for high precision agriculture projects and has been studied from different perspectives.

On a coarse level, row detection methods can be applied to detect the positions of the crop row in fields [29, 53, 54, 55]. Additional processing steps are required to detect the position of a plant along the crop row.

Furthermore, the centroid or center of a segmented plant can be determined, post-processed and used as plant position estimate [31, 56, 57, 58, 59]. These methods require a good initial segmentation of plants, situations with overlap make position estimation difficult and result in errors with these methods.

Moreover, technologies other than 2D image processing are applicable: External georeferencing can be used to re-detect the plant position in the field [60] using plant mapping. High precision RTK GPS positions of the seeds are recorded during sowing and using the recorded coordinates plants can be located later in the field [44, 45, 61, 46]. RTK GPS methods have improved, but the required sub-centimeter accuracies are not reachable yet and additional errors from seed displacement can not be corrected by seed mapping approaches.

Finally, 3D sensing and processing has been applied to the problem [62, 63, 64, 65, 66, 67]. However, these methods require special equipment (for example stereo or time-of-flight 3D cameras) and are tailored towards specific use cases.

The few existing work on plant detection and position estimation lack the ability to estimate the position of both crop and weed plants in outdoor field images. Most methods only work with large plants and additionally produce insufficient results when plants overlap because error-prone plant segmentation is performed. Seed mapping-based approaches are not applicable to precision agriculture tasks like weed control because the plant position map does not contain weed positions.

With precise knowledge of both the plant class and position, plant specific treatment can be realized. This includes plant specific weed control, selective watering or fertilizing, thinning of crops and pruning of parts of plants.

1.3 Contributions of the Thesis

This thesis develops a new machine vision approach to plant classification and plant position estimation for agricultural robots. The complete machine vision system must be suited for integration into field robots. Additionally, it must be applicable to different precision agriculture projects where information like plant class, plant position or other plant related metrics are required.

In contrast to previous studies which mostly focus on large crops such as corn or sugarbeet, the goal of this thesis is to solve these vision tasks in early stages of vegetable farming. The field situation is challenging since crop plants are very small while weed plants occur in different sizes. Simply moving the camera closer is not a solution. Moreover, plants grow close together (intra-row distance of approximately 1–2 cm) and overlap between plants occurs. The envisioned precision agriculture approaches require high precision per-pixel plant classification and sub-centimeter accurate plant position estimation. Otherwise, plants cannot be classified and treated individually and for example precision weed control is not possible.

The properties of the novel plant classification approach which fulfills these challenging requirements are as follows:

- The developed plant classification system works in real world field situations of commercial carrot farms: Inter- and intra-row weed is discriminated from crop, weed that grows close-to-crop as well as overlap of plants are successfully handled.
- The method for plant classification requires no prior plant or leaf segmentation: Extraction of shape and statistical features is performed on overlapping image patches. Then for each patch a plant classification score is determined using a supervised classification algorithm and associated to the keypoint where the patch was extracted.
- The classification results per keypoint are spatially smoothed using a Conditional Random Field (CRF) and interpolated to full image resolution via nearest-neighbor interpolation. The system outputs a crop/weed estimate for all vegetation pixels in the image. The precision loss of cell-based methods that classify large non-overlapping cells (see related work) is avoided. This novel divide and conquer approach enables the system to handle overlap of plants and region where leaf or plant segmentation algorithms struggle (for example irregular shaped leaves).
- Additionally, the classification system can be trained to discriminate more than two classes: For example a crop and multiple specific weed classes can be defined; this enables applications where for example occurrences of a special weed need to be monitored or this weed needs to be treated with a specific method.

The new method for plant detection and sub-centimeter accurate position estimation in multispectral field images has the following properties:

- Plant position estimation is formulated as a detection problem. A modified sliding window is applied and each image patch representing the local neighborhood in the image is classified whether it displays a stem or non-stem region. The resulting stem scores are smoothed and processed with non-maximum suppression to yield the estimated plant stem positions.
- During pipeline training a special procedure is conducted to generate stem an non stem patches. The characteristic appearance of the plant stem region is described with novel statistical and geometric features. The feature representation and labels

are used to train a classifier which is able to discriminate between patches displaying a stem or no stem.

- Only downward looking multispectral images of plants are processed, no additional information (like for example row location or all crop positions) is required. The stem detection process is not based on pre-segmented plant regions or pixel-based classification.
- The novel plant position estimation pipeline copes well with real world field situations, where plants overlap. It detects both crop and weed plants and therefore enables to precision agriculture tasks like single plant weed control.

After the evaluation of the individual pipelines, the estimated plant positions are merged with the plant classification image. This combined system has the following properties:

- By combining the plant classification image and estimated plant positions, crop and weed plant positions in the field are determined. Now, single plant weed treatment can be realized: For estimated plant positions where the plant classification predicts weed as plant class, a treatment with a weed removal tool is scheduled. Detected crops are skipped and therefore preserved.
- Finally, the developed pipelines and camera system are validated in a field robot in a publicly funded project: The field robot Bonirob V2 is built and equipped with the combined system developed here as well as a mechanical weed control module. Using this setup crop/weed discrimination and weed control are demonstrated in a commercial organic carrot farm in a fully automated manner.

Parts of this thesis have been published in papers [68], [69], [70] and a book chapter [71]. Furthermore, the results of the thesis have been successfully applied in a publicly funded project. In that context additional papers were co-authored: [72], [73], [74], and [75].

Additionally, a dataset of field images with intra-row and close-to-crop weed infestation was made publicly available (`https://github.com/cwfid/dataset`) to the research community with the publication [70]. The dataset includes images, ground truth crop/weed annotations as well as results from a plant classifier.

1.4 Structure of the Thesis

This thesis is structured as follows:

Chapter 2 introduces fundamental computer vision and machine learning principles. This includes the definition of features, feature extraction, classification and regression. Additionally, the Random Forest classification algorithm which is used in the thesis is presented. These concepts are used throughout the thesis and are applied and extended for plant classification and detection with field robots.

Chapter 3 focuses on the acquisition of field images which are best suited for plant and stem classification. A multispectral camera setup which delivers color and near-infrared (NIR) images is derived from requirements after comparison of available sensing principles. The second part of the chapter presents the developed vegetation segmentation method based on the Normalized Difference Vegetation Index (NDVI) and an improved filtering and thresholding scheme. After vegetation segmentation all background non-biomass pixels are masked.

Chapter 4 presents the developed plant classification system. The system processes the multispectral vegetation segmented field images to produce plant classification information. The novel plant classification system extracts features from overlapping patches, then applies a supervised Random Forest classifier, smoothing with a Conditional Random Field and interpolation to yield an estimated plant class value is assigned to each vegetation pixel. This full plant classification image is a suitable input to the introduced precision agriculture methods like plant specific weed control.

Chapter 5 develops the plant stem detection and position estimation process. The input to the plant position estimation system are only the vegetation segmented field images from Chapter 3. The system uses a novel feature extraction and classification scheme together with filtering and non-maximum suppression to output estimated plant positions. The stem detection and position estimation processes are plant type independent and produce estimated positions for all plants (both crop and weed). The plant stem position output allows plant specific precision agriculture.

Chapter 6 presents experimental results and a discussion. First, the acquisition of the datasets using a custom built field robot and their properties are introduced. Second, the plant classification method from Chapter 4 and the stem detection and position estimation method from Chapter 5 are analyzed given the datasets and their results are discussed. Finally, the combined system for plant classification and position estimation is presented and evaluated in a robotic single plant weed control task. This concludes the discussion with the farmer's perspective.

Chapter 7 summarizes the thesis and presents conclusions.

Figure 1.1 displays the structure of the thesis in a graphical form. It displays the image acquisition, vegetation segmentation, plant classification and position estimation modules together with example images and the data flow between the steps. Evaluation of results and discussion is performed for each pipeline separately and furthermore for the combined system. Finally, the output for single plant weed control is depicted with the developed and built field robot.

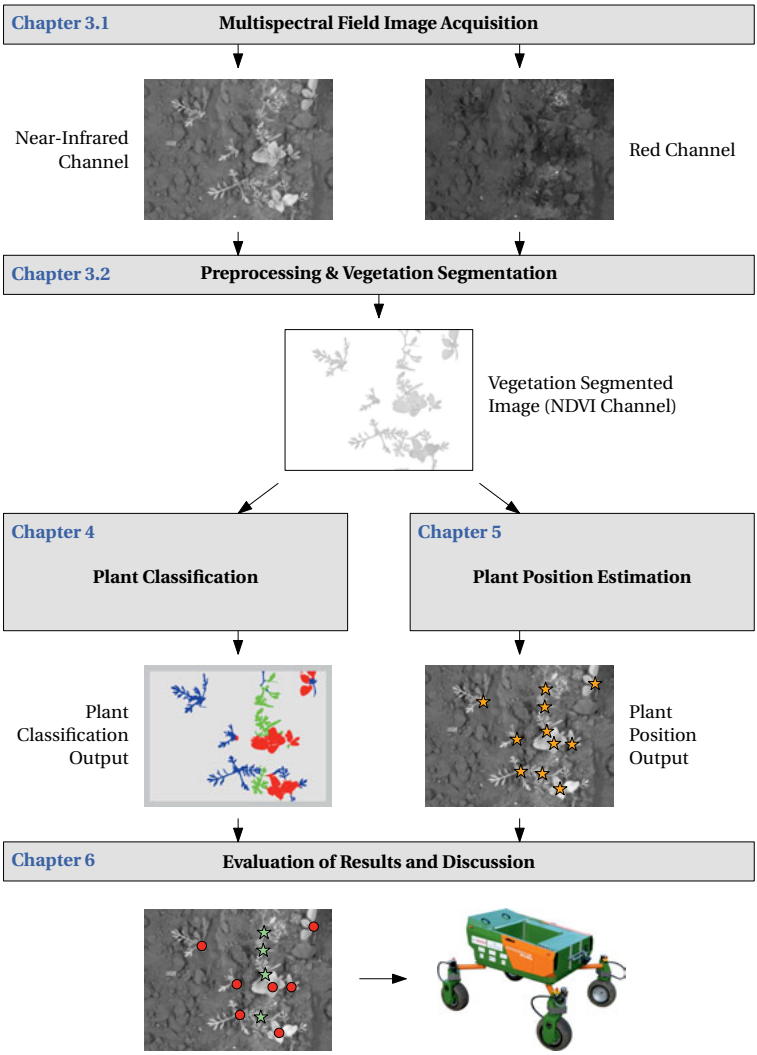


Figure 1.1: Visual structure of the thesis: Image acquisition and vegetation segmentation are the first steps. Plant classification and plant position estimation run in parallel. The evaluation and discussion is performed for each pipeline as well as for a combination. The combined output is applied with a field robot to implement single plant weed control.

2 Background: Computer Vision and Machine Learning

As stated in the introductory chapter the aim of this thesis is to automate perception and classification tasks in the agricultural domain. Today these tasks are still carried out by workers using human visual perception and cognition.

Over decades the study of the human visual system together with progress in mathematics, electronics and computer science led to the development of the field of computer vision. It comprises methods and algorithms which allow computers to acquire and process visual information. During the same time, machine learning evolved where machines are programmed with the goal to enable them to "learn".

Combining computer vision and machine learning techniques to solve perception and reasoning tasks is an active research field with a manifold of applications in the real world, the aim of this thesis included.

In the following, this chapter introduces fundamental computer vision concepts as well as machine learning techniques that are of relevance for this thesis.

2.1 Computer Vision

Computer vision is a field in computer science which studies methods and algorithms that allow computers to acquire and process visual information. It is a very active and broad field of research with many subtopics and applications in the real world.

Computer vision algorithms and their application are omnipresent: They are applied in factories where computers analyze products for defects and control robotic arms. In agriculture and food production computer vision helps guide tractors across fields, to sort and grade produce and to detect diseases. In cars computer vision algorithms detect possible accidents and issue emergency braking signals, they detect sleepy drivers and in the future will play a major role in autonomous driving. Last but not least, consumers rely on computer vision techniques; for example when using smartphone applications like bar code and QR-code readers, when they acquire and process images and videos, or upload them to image processing applications on the internet.

The discussion in the remainder of this section starts with image acquisition and representation, then low-level techniques, feature extraction and high-level techniques are introduced.

2.1.1 Digital Image Acquisition and Representation

A basic building block of computer vision systems is the acquisition and representation of image or video data. Optical image sensors or digitization equipment can be used to capture image data. Alternatively, images can be read from digital sources like storage media or the internet.

To work with image data, computers must be able to represent the visual data in a format which can be easily stored and processed: The most common representation for digital images is a spatially discretized data structure which can be expressed as a two dimensional array (matrix). Each element in the matrix comprises a pixel in the image and for each pixel a scalar (p) or vector ($\boldsymbol{p}$) is stored to encode the intensity or color information.

A common notation for an image (I) is the representation with a row-major h-by-w matrix. The parameter h defines the height and the parameter w the width of the image in pixels. A single pixels $p_{u,v}$ is addressed in the image matrix at coordinates u, v, where the origin of the u, v-axes is in the top left corner of the image (see Figure 2.1).

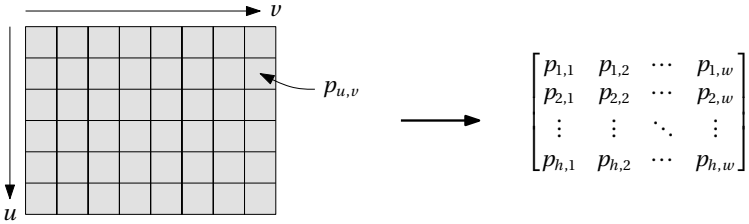


Figure 2.1: Coordinates in image space and matrix representation.

Additionally, color must be represented in the image. The most basic notation is a grayscale image where each pixel p encodes a single brightness value. Typically, a single Byte is used to encode 256 different brightness values from black (0) to white (255).

To represent color, a so-called color space is used. For digital images the Red, Green and Blue (RGB) additive color space is common. It encodes color using a vector with three entries: $\boldsymbol{p}_{\mathrm{u,v}} = [r, g, b]$ where r, g and b encode the intensity of red, green and blue respectively. Other color spaces exist and can be used depending on the acquisition device and the specific application: For example the Cyan, Magenta, Yellow and Black (CMYK) subtractive color space (mainly used in the printing industry) or the HSL color space which represents Hue, Saturation and Lightness (HSL) with 3 values.

So far only colors of the visual spectrum are represented. However, some applications can use information from the near-infrared (NIR) or ultra-violet (UV) part of the spectrum. Therefore extended color representations can be defined: In addition to the RGB channels, an image is then composed of 4 or more channels. Such images are called multispectral, for example an RGB plus NIR image with 4 channels. If this concept is extended further and many channels are used such images are called hyperspectral.

In addition to color information also depth information (distance from sensor) can be acquired and stored for each pixel: Such images can be produced by stereo camera systems or cameras with time-of-flight sensors. The resulting images with depth information are sometimes called 2.5D images, because the u, v coordinates are discrete (arranged as defined by the fixed image grid) and therefore no occlusions in 3D space can be represented. For example only the front side of a cube can be modeled.

Full 3D data on the contrary can represent depth occlusions. Such full 3D data consisting of a list of points in 3D space (represented for example by metric x, y, z coordinates) and optionally additional data (for example a vector describing color or surface normals) are called point clouds. Special algorithms and methods form the separate field of point cloud data processing which is beyond the scope of this study.

2.1.2 Preprocessing / Low-level Techniques

Once a digital image or image sequence is available, low-level techniques can be applied to improve image quality or to preprocess the image in various aspects.

Color correction is a frequently applied low-level step. All color vectors in the image undergo a certain transformation to for example enhance contrast, remove overexposure, to adjust white balance or to transform a color image into a grayscale image.

A special low-level color processing step is the removal of the Bayer pattern. Digital color cameras often just sense one color (using a color filter) for each physical pixel. An array of four pixels with red, green, green and blue color filters (called Bayer pattern) is used [76], then interpolation yields four full RGB pixel values from the four single color pixel values which the camera physically captures.

Cropping, scaling, resizing, rotation or distortion correction are other low-level steps which can be applied to images to improve the image, cut out the area of interest or to reduce distortion effects of the lens through camera calibration [77].

Another common preprocessing step is filtering: For example noise reduction filters, smoothing with Gaussian filters or sharpening of images. Low or high pass filters can be applied to find edges or continuous areas. Filters are commonly realized by convolving a filter kernel over the image.

Segmentation can be used to split an image into multiple objects/regions [78]. This is for example applied during the production of videos which are filmed in front of green screen. Using segmentation algorithms which mask the green pixels, the background can be replaced by for example a landscape view. Simple approaches utilize pixel-wise grayscale/color thresholds [79] or edge information. More complex algorithms apply for example graphs [80] or contours and textures [81] to segment the image.

Most preprocessing techniques operate on single pixels, a window of pixels (filtering with high or low pass filters) or the whole image. The output of the preprocessing steps is again an image with potentially a different color space, resolution, segmented parts, etc.

2.1.3 Features and Feature Extraction

Feature extraction is a basic building block of many computer vision algorithms. The idea behind feature extraction is to generate a representation for data (for example an image or an object contained in an image) which describes the original data using a more abstract representation with less redundancy. Therefore, feature extraction can be seen as dimensionality reduction technique.

A desired property for a feature representation is that the derived feature retains the relevant aspect of the original data point. Often invariance to transformation in the original dimension (for example rotation of an object in the image) is a desired property of a feature description.

Examples for features which can be applied to any data are Principal Component Analysis (PCA) [82], kernel PCA, the calculation of histograms or the usage of embeddings (i.e. projections which retain a notion of distance) [83].

In the image domain many well-known and established features and feature detectors exist. Basic local features are blobs, edges, corners or regions in the image [84]. Additionally, there exist features which describe directions of edges or gradients [85]. Contour based features describe shapes of objects for example using snakes or contours.

Furthermore, the location in the image can also be leveraged as feature, because objects might occur predominantly in specific parts of the image or in a specific relation to other objects in images.

The transformations to which the image features should be invariant include but are not limited to the choice of camera system, changes in viewpoint, focal length, illumination. Additionally invariance to change of scale, rotation and translation of the image and projective changes are desired properties of features for computer vision tasks.

Most advanced feature extraction techniques for image processing can be decomposed into a two step process:

1. In the *keypoint detection* step, the area of interest is selected: This can be a point, a segmented object, a point with a neighborhood, a bounding box, etc. In the literature also the term interest point is used as synonym for keypoint.
2. The *descriptor extraction* step derives the numerical description — called feature vector — which describes the area of interest in a descriptive form which at best is invariant to transformation and has no redundancy.

An example for a well known state of the art feature is the Scale Invariant Feature Transform (SIFT) [86]. Given an image, the keypoint detection selects points (maxima in a linear scale space) in the image and calculates scale and orientation information for each keypoint. Using a keypoint with its orientation and scale information, the descriptor is calculated which describes the local image region around the keypoint. SIFT features are invariant to scale and shift transformations as well as rotation in the image plane. Variants have

been proposed to make SIFT also invariant to full affine transformation including larger changes of viewpoint [87] or to improve the extraction speed [88].

2.1.4 High-level Computer Vision

High-level computer vision algorithms solve high-level tasks like detection, reconstruction or classification using image data as information source. Most high-level algorithms are complex combinations of several building blocks and are highly task dependent. In the following a few generic high-level tasks are described:

An example for a generic high-level computer vision task is detection and tracking of objects in images [89]. For example cars and pedestrians are to be detected in a stream of images captured by a driving car and then using subsequent images a trajectory of the objects is estimated.

Another example is a Visual Simultaneous Localization And Mapping (VSLAM) [90, 91]. Using camera images feature points are extracted and their position in 3D space is estimated. Then using subsequent images the motion of the camera is estimated together with a map (for example all 3D feature points) of the environment.

Overall, a large variety of such high-level computer vision problems exits. The goal of this thesis is to develop classification and detection methods for agricultural images which are introduced in depth in Chapters 4 and 5.

Many computer vision techniques also apply learning methods which allow the computer vision system to learn form data using machine learning — which is introduced in the next section.

2.2 Machine Learning

Machine learning is a discipline in computer science that studies principles and algorithms which are able to learn from data. The learning process involves the deduction of a model from the input data the algorithm was given. Subsequently, this model can be used to infer decisions or make predictions given new data samples the model has not seen before.

In addition to the study of algorithm which are capable of training models based on data and subsequently making predictions, important parts of machine learning are data handling for training and testing, preprocessing and evaluation of results using suitable metrics. All these aspects are introduced in the remainder of this section.

2.2.1 Training and Application

Generally, machine learning processes can be divided into different stages: During the first stage called *training*, the algorithm is presented with data and constructs a model. Second,

during the optional *test* phase, the model is tested to estimate how well it performs on data which has not been used during training. Third, in the *application* phase, the previously trained model is used to predict information for new input data.

Data and Labels Data is very important for machine learning: On the one hand, training data is required in order to allow the algorithm to build its model. On the other hand, test data is required to evaluate how the trained machine learned model performs. For most machine learning approaches, the data must be labeled. The term *label* implies that for each instance of data (for example a temperature measurement) a label (for example 'winter' or 'summer') is available. The type of the label can be a categorical values, a continuous value or complex types like a list of values. Section 2.2.2 below introduces advanced machine learning modes where no or partially labeled data is sufficient.

The data on which the machine learned model is applied normally does not have labels, the algorithm is used to estimate a label. The estimated labels are the output of the machine learning process.

Cross-validation A common approach of combined training and testing while using labeled data as efficiently as possible is cross-validation [92]: In n-fold cross-validation the labeled data is split into n folds. Then a single fold is used as test data while all other folds are used for training. This is repeated until every fold has been used as test data once. The n classifiers are evaluated on the respective test data sets. To achieve an aggregated score, the classification scores of each fold are aggregated by summing before classification metrics are calculated (see below).

Cross-validation increases training time compared to a simple split of the data in test and training set. With cross-validation however all data is used during training and thus generally better performance and therefore for example better parameter selection can be achieved.

When a suitable machine learning algorithm and its parameters are found for example using cross-validation, the whole labeled dataset can be used to train a final model. This model can then be applied in the application phase.

Notation In the following, a notation is defined which is used throughout the thesis for machine learning models and the data items involved.

Input data is expressed as matrix $\boldsymbol{X}$ where rows represent single data instances. Such a row is also called feature vector $\boldsymbol{f}$. The elements f_i of the feature vector $\boldsymbol{f}$ are individual features (numerical or categorical). The labels are expressed as vector $\boldsymbol{l}$ where the label l_i corresponds to the i-th data instance (i-th row in matrix $\boldsymbol{X}$). Additionally, the symbol $\boldsymbol{g}$ shall denote ground truth labels which are for example used during training or when evaluating the output labels $\boldsymbol{l}$ to judge the performance of the machine learning model (see upcoming Section 2.2.4).

The training and application process can now be written with this notation:

$$\mathrm{F}_{\mathrm{train}}\left(\boldsymbol{X}_{\mathrm{train}}, \boldsymbol{g}, \phi\right) \longrightarrow \mathrm{Model} \tag{2.1}$$

$$\mathrm{F}_{\mathrm{application}}\left(\boldsymbol{X}_{\mathrm{application}}, \mathrm{Model}\right) \longrightarrow \boldsymbol{l} \tag{2.2}$$

The functions F_{train} and $F_{application}$ are algorithm specific functions. All parameters of the machine learning algorithm are expressed by ϕ in Equation (2.1). The estimated labels $\boldsymbol{l}$ are the desired output after processing the application data $\boldsymbol{X}_{application}$.

Some classifiers are not only able to estimate the most likely class label l, but also supply a certainty score vector $\boldsymbol{s}$ for each classified data instance. The score vector contains a relative score for each of the possible classes. In the following we always assume that each score vector $\boldsymbol{s}$ is normalized, i.e. $\sum s_i = 1$.

2.2.2 Learning Modes

Depending on the availability of labeled data, machine learning problems can be discriminated into three major modes: Supervised learning, unsupervised learning and semi-supervised learning.

In *supervised learning*, all data instances which are used during training are labeled with ground truth labels. A typical use case is classification where the labels are categories forming the different classes the classifier must discriminate.

Unsupervised learning covers the use case where all data instances are unlabeled. The application of machine learning in such problems can be for example clustering. In clustering the algorithm tries to group the instances according to a distance measure in feature space which allows grouping of instances.

Semi-supervised learning approaches can work with partially labeled training data: When only a fraction of the data is labeled, the semi-supervised algorithms make use of the unlabeled instances during training. This approach can have benefits compared to discarding unlabeled data and applying a standard supervised approach.

In addition to these three basic learning modes further specialized machine learning techniques exist: Reinforcement learning [93] covers the learning of policies given either a positive or negative reward for each data instance. Active learning [94] can actively select training instances which should be labeled to train a classifier while minimizing the total amount of labels required. Transfer learning [95] can utilize data from a different domain to improve model training in another similar domain.

The machine learning community is an active research field and new learning modes evolve, however, supervised and unsupervised learning remain the most important and most widely applied approaches.

2.2.3 Classification and Regression

Classical machine learning tasks can be discriminated into the two broad categories of classification and regression [96]. Classification problems try to classify data instances into different categories whereas regression problems try to infer a continuous variable.

The goal of *classification* is to associate data instances to different categorical labels. For example the classification of patients into the two classes infected or non-infected given the output of a blood test. The output of classification tasks is a categorical variable from a set of predefined options. Besides simple binary classification tasks (the category to be estimated consists of two different options/labels), in so-called multi-class classification tasks the output category consists of 3 or more options.

The aim of *regression* is to estimate a continuous quantity. For example the price of a house given its size and construction year. The output of regression tasks is a continuous variable or vector (multidimensional regression).

Many machine learning algorithms can perform both classification and regression when implemented accordingly. Before concrete algorithms are discussed, first the the next section studies the how machine learning processes can be evaluated.

2.2.4 Evaluation of Machine Learning Processes

An important aspect of machine learning research is the evaluation of machine learned models and the optimization thereof. The datasets analyzed are often large and manual inspection of each estimated instance of data is not feasible.

The performance of the classification can be analyzed by using metrics and ground truth data. A metric compares the estimated values l from Equation (2.2) with the ground truth values g from Equation (2.1) and yields a performance value. In addition to metrics graphical tools like the Receiver Operating Characteristic can be applied.

Confusion Matrix Basic metrics are the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). These metrics can be arranged in a confusion matrix (see Figure 2.2) which also explains how the values are calculated by comparing ground truth and estimated values. For multi-class classification the confusion matrix is augmented with more rows and columns.

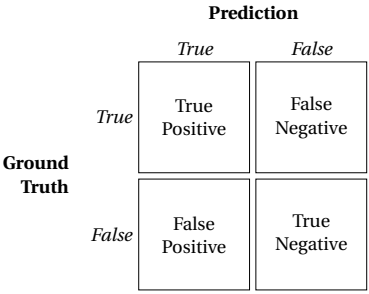


Figure 2.2: Layout of a general confusion matrix.

Classification Metrics The widely-used classification performance metrics average accuracy, precision, recall and F1-score are derived from the basic classification measures. For binary classification these metrics are defined as follows:

$$\text{average accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fn} + \text{fp} + \text{tn}} \tag{2.3}$$

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{2.4}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{2.5}$$

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.6}$$

To evaluate multi-class classification similar metrics can be defined. The multi-class average accuracy metric is defined as the average of the per class average accuracies:

$$\text{average accuracy}_{\text{multi-class}} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\text{tp}_i + \text{tn}_i}{\text{tp}_i + \text{fn}_i + \text{fp}_i + \text{tn}_i} \tag{2.7}$$

When constructing the metrics precision, recall and F1-score, different forms of averaging can be used in the multi-class case: In macro averaging [97] (indicated by subscript M in the formulas) averaging is done over the per class scores. The symbol N denotes the total number of classes and the index $i \in (1, N)$ indexes a specific class.

$$\text{precision}_M = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\text{tp}_i}{\text{tp}_i + \text{fp}_i} \tag{2.8}$$

$$\text{recall}_M = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\text{tp}_i}{\text{tp}_i + \text{fn}_i} \tag{2.9}$$

$$\text{F1-score}_M = \frac{2 \cdot \text{precision}_M \cdot \text{recall}_M}{\text{precision}_M + \text{recall}_M} \tag{2.10}$$

In addition to macro averaging, also micro averaging can be applied (indicated by subscript μ in the formulas). There the average is built over all samples and not the per class weighted average.

$$\text{precision}_\mu = \frac{\sum_{i=1}^{N} \text{tp}_i}{\sum_{i=1}^{N} \text{tp}_i + \text{fp}_i} \tag{2.11}$$

$$\text{recall}_\mu = \frac{\sum_{i=1}^{N} \text{tp}_i}{\sum_{i=1}^{N} \text{tp}_i + \text{fn}_i} \tag{2.12}$$

$$\text{F1-score}_\mu = \frac{2 \cdot \text{precision}_\mu \cdot \text{recall}_\mu}{\text{precision}_\mu + \text{recall}_\mu} \tag{2.13}$$

The advantage of macro averaging is that all classes are treated equally even when the number of data instances for one class is very small. When micro averaging is used in that case, a poor or good performance of the small class is under weighted in the metric.

ROC Curve If the output of a classifier is a continuous confidence estimate (for example score vector $\boldsymbol{s}$) in the range from 0 to 1, a decision threshold is selected to derive the final label. The threshold can be chosen freely and has an impact on the performance. A so-called Receiver Operating Characteristic (ROC) [98] curve plots the true positive rate over the false positive rate for all decision thresholds, see Figure 2.3. The closer the ROC curve extends towards the upper left corner, the better the classification results are. A random classification choice yields a diagonal curve.

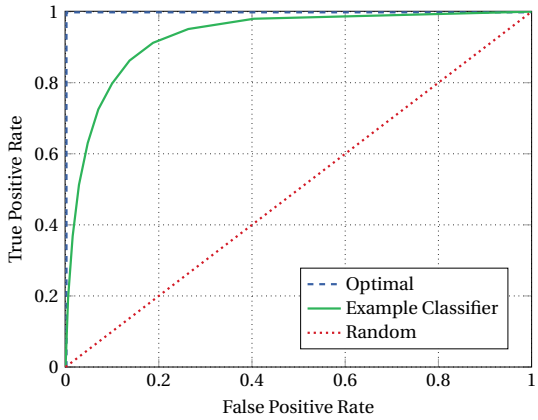


Figure 2.3: Exemplary Receiver Operating Characteristic (ROC) curve.

In order to evaluate multi-class classification problems with ROC curves aggregation is necessary. In an N-class problem N one vs. all ROC curves can be plotted: To create the ROC curve for class i, a binary ROC curve is created by comparing class i against a pseudo class built by aggregating all other classes. Another option is to plot one vs. one ROC curves. This however results in $N^2 - N$ curves and their interpretation in more difficult.

2.2.5 Machine Learning Algorithms

Well known machine algorithms which support both classification and regression tasks are for example Support Vector Machines (SVM) [99], Boosting classifiers like AdaBoost [100, 101], Neural Networks [102, 103, 104], Decision Trees [105] or Random Forests [106].

All machine learning algorithms have individual weaknesses and benefits and the choice of algorithm is application dependent. In the following the SVM, Decision Trees and the Random Forest algorithm are introduced in detail.

Support Vector Machine The SVM performs classification or regression by applying a hyperplane to separate the data instances into two classes (positive and negative). The selection of the hyperplane is done such that the distance of the nearest training data points (called support vectors) from the hyperplane is as large as possible [99]. Therefore, the SVM is also called a maximum margin classifier.

In its most basic form the SVM is a linear algorithm which is able to process linearly separable data. Using the so-called kernel trick [107] the data instances are transformed into a higher dimensional space where they are separated using a hyperplane. In the original feature space the projected hyperplane forms the desired non-linear decision boundary which separates the training data. Since not in all cases a linear separation even in higher dimensional space can be achieved, SVMs were extended to for example soft margin variants. There, some outliers on the wrong side of the hyperplane are accepted to avoid overfitting the classifier to noisy data.

A drawback of SVMs is that by construction they only supports two classes. For more classes for example multiple binary classifiers must be trained. Furthermore, individual features must be normalized to similar magnitude to avoid prioritizing some features.

Decision Trees Decision trees are a well known machine learning algorithm for classification and regression. A decision tree is built using binary decision nodes and leaf nodes which are arranged in a tree structure. A decision node is defined by a binary true/false decision. Depending on the comparison result, the binary tree is either followed to left or right child node in if − then − else fashion. After a series of decision nodes a leaf node is reached. Leaf nodes represent the result of the decision (a single class value in the classification case) and once a leaf node is reached the leaf node's result is returned.

In most implementations the binary decision equals an axis-aligned partitioning in feature space: A single feature f_i of the feature vector $\boldsymbol{f}$ is compared to a threshold. The specific nature of the binary decision can be adapted to the use case or data: Variants with linear splits or more complex decisions have been proposed. However they introduce more complexity and often the simple single feature criterion is chosen.

Figure 2.4 displays an example decision tree with 3 decision nodes, 4 leaf nodes and in total 2 different output classes (crop or weed).

The training of such a decision tree can follow different rules. The most relevant step of the decision tree training process is how the splitting criterium of each decision node is determined.

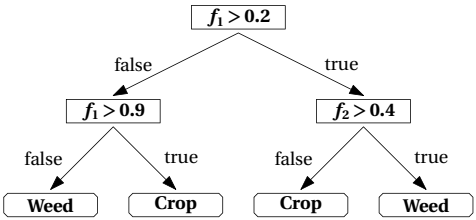


Figure 2.4: Simple binary decision tree. The square boxes represent decision nodes, where one feature is checked against a threshold and a true/false decision is made. The terminal nodes are also called leaf nodes and represent the decision output of the tree (in this example weed or crop).

During training of a splitting node the input is the current list of feature vector and label tuples $[(\boldsymbol{f}, l)]$. The output of the training step is a binary splitting criterion (selected feature and threshold) plus a list of feature vector and label tuples for the left branch $[(\boldsymbol{f}, l)]_{\text{left}}$ and the right branch $[(\boldsymbol{f}, l)]_{\text{right}}$. From thereon, splits are performed recursively until all the labels in the list are pure, i.e. all l belong to just one class. Then splitting is stopped and a terminal node with class l is inserted.

The process of selecting the binary splitting criterion consists of selecting a feature and a threshold. First, all features are considered as possible splitting criteria. Second, for each feature all possible thresholds which results in different splits are possible candidates. All possible thresholds can be determined by sorting the selected feature values in the current data and taking all unique values as possible thresholds. Now, a cost function is required to select the best splitting criterion from all these possible splits.

A cost function which is often chosen to judge splitting quality is the Gini impurity: It quantifies how pure a given split into two lists is. The Gini impurity is calculated as follows. Consider again a problem with N classes with i indexing these classes $i = 1, \ldots, N$ and a list of labels l for which the impurity is to be calculated. For each class the relative occurrence of labels p_i in the list of labels l is calculated: $\boldsymbol{p} = [p_1, \ldots, p_N]$. Then the Gini impurity is calculated:

$$G(\boldsymbol{p}) = 1 - \sum_{i=1}^{N} p_i^2 \tag{2.14}$$

For example in a two class setting for a pure list of labels with $\boldsymbol{p} = [0, 1]$ the Gini impurity is 0. For a completely mixed list of labels with $\boldsymbol{p} = [0.5, 0.5]$ the Gini impurity is 0.5. For all other label distributions the Gini impurity is in-between these values. Therefore splits with lower Gini impurity are preferred when constructing the decision tree.

Now back to the selection process of the splitting criterium of a decision node: All possible splits are evaluated by calculating the cost consisting of the sum of the Gini impurity

for both the left and right list of labels ($[l]_{\text{left}}$ and $[l]_{\text{right}}$). The splitting criterion with the lowest cost is chosen. From thereon splitting is recursively performed on the left and right branches until all splits result in pure terminal nodes.

Decision trees however tend to overfit the training data they are trained with. Many extensions have been proposed to improve decision trees by for example pruning [108] or not training trees to fully pure leaf nodes. The most promising concept however is the Random Forest which is introduced in the next paragraphs.

The Random Forest Algorithm The Random Forest algorithm is an established and widely used machine learning method which extends on decision trees and was initially proposed by Breiman [106]. Random Forests use multiple decision trees to form a so-called forest. They have been applied successfully in many domains including machine vision [109, 110], bioinformatics [111], remote sensing [112], robotics [113] and others [114].

Random Forests are a class of algorithms which train multiple tree like classifiers while applying randomness to modify the training set and/or tree training approach. The original Random Forest algorithm by Breiman trains an ensemble of modified decision trees making use of randomness for feature selection and through bagging [106]. This yields best results when compared with other methods of randomness which can be applied [115].

The first component which utilizes randomness is random feature selection during training. During splitting node training only a random subset of features is considered. Then the splitting node is constructed by selecting the most discriminative feature from the subset using for example the Gini impurity. Bagging is an ensemble method where a set of classifiers is trained using smaller bootstrapped training sets. In Breiman's Random Forest algorithm random sampling with replacement is performed to create the bootstrapped training sets for each tree. Bagging is the second component in Random Forests which applies randomness.

The output of each tree is a vote for a specific class, these votes are then aggregated to an overall majority vote. If classifier scores are desired the votes are collected into a score vector $\boldsymbol{s}$. Each tree votes for a specific class and the respective score s_i is incremented. The score vector (sometimes also called vote vector) is typically normalized to sum to 1. It can then be used to plot a ROC curve and to determine an optimal use case dependent threshold that yields the final estimated class.

A major advantage of Random Forests is the availability of out-of-bag errors. During construction of the bootstrapped training data sample several training data instances are not sampled. These instances create the out-of-bag data which can be used as test data for this specific tree. This allows an evaluation of the trees performance without a special test set. The out-of-bag errors can be aggregated over the whole forest to get an overall estimate which performs comparable to cross-validation [116]. The out-of-bag error can be evaluated continuously and used as criterium when training of additional trees can be stopped. Figure 2.5 gives a graphical overview of the training process of a Random Forest and the out-of-bag error estimation.

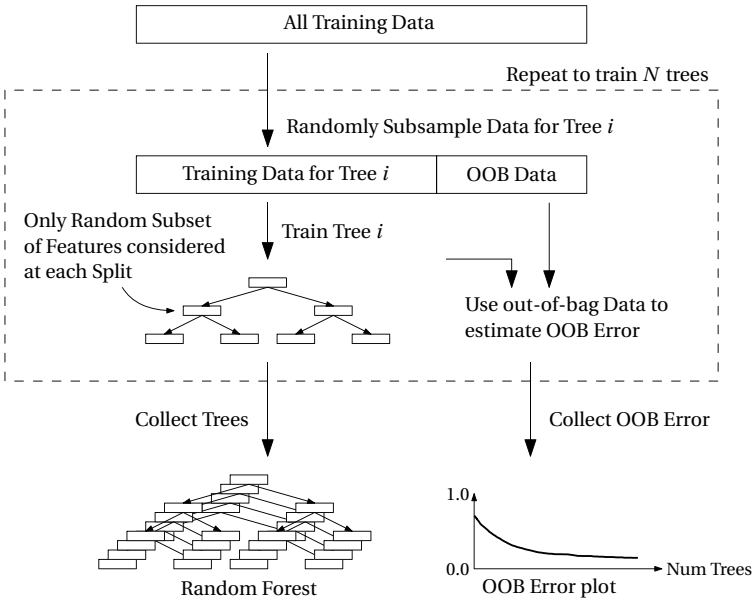


Figure 2.5: Training of a Random Forest and out-of-bag error calculation.

The Random Forest classifier's strengths are that only few parameters need to be adjusted [117] and a major benefit over plain decision trees is that Random Forests mitigate overfitting [106]. Additionally, the algorithm can be easily parallelized which allows high performance implementations on multi-core processors, adapted to GPUs or coded for special purpose hardware such as ASICs or FPGAs [118].

The main parameters of a Random Forest are

- the number of decision trees which are grown during training. In practice a high number of trees should be trained as more trees improve classification accuracy. The concrete number is often selected by analyzing at the out-of-bag error and choosing a number of trees where the out-of-bag error starts to converge towards its minimum.
- the number of features which are considered at each splitting node. Typically, this parameter is set to $\lfloor \sqrt{n_{\text{features}}} \rfloor$.
- the depth until which the trees are trained. This is expressed in the minimal number of data instances which are stored in each leaf node. Typically, trees are trained until all leaf nodes are pure and trees are not pruned.

Additionally, there exist variations of Random Forests which can be chosen: For example the specific implementation of a decision tree and which kind of splitting function it applies

internally can differ. There exist decision trees which apply a single axis aligned split, other apply linear or quadratic functions to partition the feature space [119]. In practice axis aligned splits (i.e. using one feature as criterion to split the data into left and right) are used almost exclusively in all implementations. Furthermore, since its introduction as supervised algorithm many variants were proposed: For example cascaded forests [120] or Random Forests with online training [121].

For the remainder of the thesis the Random Forest algorithm is selected since it naturally supports multiple classes, outputs class certainty scores, is easy to interpret and has few parameters to tune.

2.3 Summary

This chapter introduces computer vision and machine learning techniques which are relevant for the remainder of the thesis:

- The field of computer vision is presented: The acquisition and representation of digital images, low-level techniques, features and feature extraction as well as high-level computer vision are explained.
- Principal concepts of machine learning are introduced: Training and application phase, classification and regression as well as the evaluation of machine learning is discussed. Finally, machine learning algorithms are introduced and SVMs, decisions trees and the Random Forest algorithm are explained in detail.

3 Multispectral Image Acquisition and Vegetation Segmentation

This chapter introduces the image acquisition methodology and the developed vegetation segmentation method. For plant classification a multispectral sensing approach is developed. The specific camera setup capable of producing such images is introduced.

Additionally, an important preprocessing step is robust removal of background pixels in the field images. Using the multispectral input image a robust approach is developed and the masked field images will be used for all further processing steps in this thesis. Figure 3.1 displays an example image.

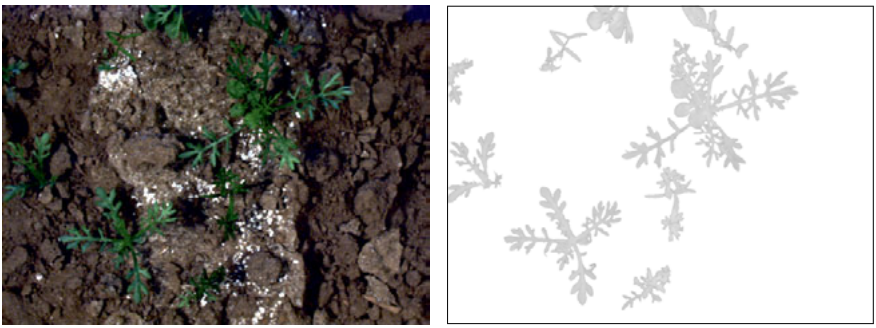


Figure 3.1: Sample image (color channel, left) and result of vegetation segmentation (right). These images are produced by the camera system and segmentation method which are discussed in this chapter.

A dataset which is acquired with the multispectral camera setup and segmented using the developed vegetation segmentation approach is also made available to the public. It is published online in conjunction with the publication [70].

3.1 Multispectral Field Image Acquisition

This section introduces the benefits of multispectral images when field images are analyzed with computer vision. Additionally, a camera setup which can deliver such images is selected and described in detail.

In the following a close-range sensing scenario is assumed, where images of a field are captured with ground-based vehicles (see Section 6.1.1 for a discussion about field robots and a specific robot developed for the task addressed here). Remote sensing on the other hand is a very established domain where images are shot from airborne or spaceborne platforms. The ground resolution such systems can achieve is a lot coarser compared to ground-based system. For this reason and because after the image analysis a direct intervention on the ground is envisioned the focus here lies on ground based sensing.

Furthermore, without restriction of generality, the camera is considered to be mounted orthogonal to the ground plane. Different mounting angels are possible and do not invalidate the findings of this chapter. However, the orthogonality assumption makes it easier to describe parameters such as the distance to the ground plane or the mean image resolution at ground plane distance.

3.1.1 Red-Edge Property of Plants

Plants exhibit a very distinct reflection property when their reflectances in the red and near-infrared (wavelength larger than 0.7 µm) bands are compared. Figure 3.2 displays the reflectance of a plant vs. soil in the visible and near-infrared wavelengths.

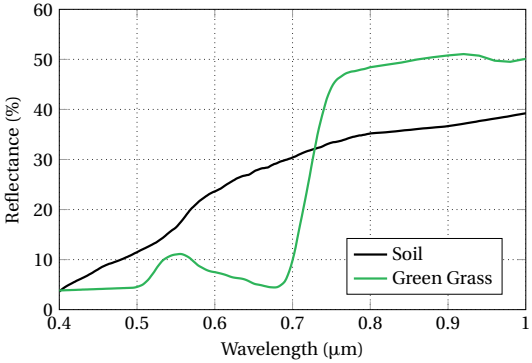


Figure 3.2: Reflectance plot of green grass and brown soil. The raw data for the plot stems from the ASTER database [122] (samples “grass” and “87P3665” for soil). Vegetation exhibits the red-edge property which is characterized by a steep increase of reflectance between the red and near-infrared bands.

Green plants absorb energy in the red band and utilize this energy in the photosynthesis process. In the near-infrared band most of the energy is reflected by the plant. The high reflectance is not present by chance, rather it functions as protection against overheating and cell damage which would be the result of absorption of the energy in wavebands not suitable for photosynthesis.

This distinct steep increase of reflectance between the red and the near-infrared band is characteristic for plants with active photosynthesis and also called the *red-edge* property. Other objects in field images like soil, stones, dead leaves etc. do not exhibit this reflectance behavior. Especially the reflectance of soil is flat in the red-edge region (see Figure 3.2).

The red-edge behavior of plants is suitable for discrimination of soil and plants. Therefore in the following a multispectral camera setup is derived which also contains a near-infrared channel in addition to the normal red, green and blue channels of standard color cameras.

3.1.2 Methods for Multispectral Image Acquisition

A variety of imaging methods or special imaging sensors can be employed to capture a multispectral image. The following major approaches exist:

Multiple Cameras In a multiple camera setup, a normal RGB camera is used together with a near-infrared camera [123, 124]. The two cameras must be rigidly mounted close to each other in order for both cameras to view the same scene. A pixel-wise registration of both images is possible in general, however in situations with occlusion such a mapping cannot be found for all pixels in the images. Downing et al. extend this to an array of 25 cell phone cameras to capture hyperspectral images, but do not present a working system yet [125].

Spinning Filter Wheel A spinning filter wheel is placed in front of the camera which is sensitive in all desired wavelengths [126, 127]. The filter wheel is sectioned into different areas which are bandpass filters for the different channels (for example red, green, blue and near-infrared). The camera captures multiple timely synchronized images (one for each filter in the wheel) which are combined into the multispectral image. When stationary scenes are captured (for example paintings in museums) the filters can also be swapped manually, instead of being arranged on a wheel [128].

Tunable Filters Tunable filters are optical elements for which the filtering function can be tuned electronically [129, 130]. Special filters exist which can be tuned to bandpass only red or near-infrared light. Such a filter can be placed in front of a camera which is sensitive in all wavelengths: Multiple images are then captured while the optical filter is tuned to bandpass only the desired wavelength band.

Interleaved Illumination of Scene This method is similar to the filter wheel setup. Instead of a passive setup with filters, the scene must be shaded and is artificially illuminated. The illumination is performed with light sources of different wavelengths and is synchronized to camera exposure times [131, 132]. Then each light source is activated for one frame and when for each wavelength a frame is recorded, all frames are combined into a single multispectral image.

Beam Splitting Setup Beam splitting cameras split the light coming through a single lens with a beam splitter [133, 134]. Then each beam hits a separate image sensor. Setups

with two or three sensors with separate bandpass filters are common. Each image sensor is designed to capture a different part of the spectrum.

Special Filter Pattern Customized Bayer-like filter patterns can be created where one near-infrared filter is placed in the pattern grid. For example a R, G, B, NIR square pattern can be used to captured near-infrared in addition to the visible channels [135]. However, de-bayering becomes more difficult and NIR information is only directly available for one fourth of all pixels.

In the following four suitable approaches are compared: 1) multiple cameras, 2) filter-based setups (both spinning filter wheels and tunable filters), 3) interleaved illumination and 4) beam splitting setup. The special Bayer pattern setup is neglected because these system are currently not commercially available.

These possible camera setups differ greatly in theoretical and practical properties: Cost, image viewpoint, acquisition timing, moving parts, complexity of sensor rig, commercial availability, robustness of sensor setup for outdoor use, etc. Table 3.1 compares the four major approaches in regard to these parameters.

The evaluation indicates that the beam splitting setup has the most promising properties for the desired use case. The beam splitting setup allows capturing of a multispectral image (RGB and NIR) in a time synchronized manner from the same viewpoint with the same field of view. Additionally, the setup comprises only a single camera body and single lens, but no other parts (filter, multiple cameras) which must be rigidly mounted and secured for outdoor use. A drawback of the beam splitting setup is the cost of such cameras. All in all, the beam splitting setup is selected and a concrete system is presented in the next section.

3.1.3 Beam Splitting Multispectral Camera System

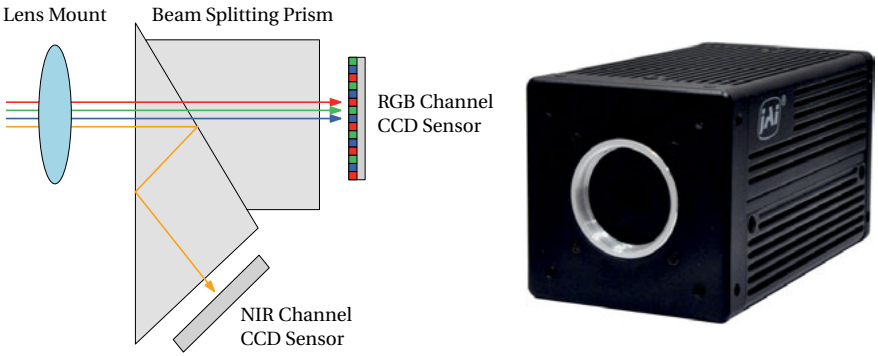
The selected camera setup is a beam splitting camera with a separate color and an near-infrared channel. The camera manufacturer Jai produces such a camera called Jai AD-130GE [136]. Figure 3.3a displays the sensing principle with the beam splitter and Figure 3.3b shows the compact housing.

The camera contains two images sensors which are aligned to the optical path to have the same field of view. The first sensor is a common RGB color sensor with a Bayer pattern, the second is a monochrome near-infrared sensor (no Bayer pattern). With a suitable lens this setup allows to capture two images at the same time of the same scene, the first being a normal color image and the second a monochrome near-infrared image.

Figure 3.4 depicts the spectral sensitivity of the JAI camera's different channels. The sensitivity of the near-infrared channels is overall lower than the color channels. This property has to be considered when setting up the camera system and for example the red and near-infrared channel's values are further processed by computer vision algorithms.

Table 3.1: Comparison of different setups for multispectral image acquisition. Positive (+), neutral (0) and negative (-) marks are used to perform a rating.

	Multiple Cameras	**Filter Based**	**Interleaved Illumination**	**Beam Splitting**
Image Viewpoint	- Different viewpoints	+ One viewpoint	+ One viewpoint	+ One viewpoint
Acquisition Timing	+ Single shot per camera (channel)	- Channels are time interleaved	- Channels are time interleaved	+ Single shot for all channels
Moving Parts	+ No	- Color wheel	+ No	+ No
Complexity	- Rigid setup and calibration needed	- Complex filter hardware needed	- Active flash-like illumination required	+ Low
Availability	- Separate parts, must be assembled	+ Yes	- Separate parts, must be assembled	+ Yes
Cost	- Multiple cameras	0 Camera plus filter	0 Camera plus lighting	- Camera with special prism
Outdoor Use	0 Large setup, mount must be rigid	0 Large setup (filter), color wheel moves	- Strong illumination required	+ Single camera body, easy setup



(a) Sensing principle

(b) Camera body

Figure 3.3: Multispectral sensing system: The Jai AD130-GE camera (b) outputs a color and near-infrared image (a). Internally this is realized using a beam splitting prism and two separate imagers. Image (a) is own work based on [136].

Now a complete camera setup is constructed (camera body with a lens) and the resulting properties are presented in Table 3.2. The choice of lens and mean distance to ground is application dependent.

Table 3.2: Parameters and properties of camera system.

Parameter	**Value**
Camera Model	JAI AD-130GE
Image Resolution	1296 × 966 px
Lens	Fujinon TF15-DA-8
Focal Length	15 mm
F-number	4
Mean Distance to Ground (d)	450 mm
Ground Resolution	~8.95 px/mm
Field of View X (at Distance d)	~145 mm
Field of View Y (at Distance d)	~108 mm

The values presented here are optimized for robot mounted acquisition of field images. The mean distance to ground (d) is defined to 450 mm and the focal length of the lens to 15 mm. This results in a resolution at ground level of approximately 8.95 px/mm. Such a high resolution is required in the application context where small fragments of plants which are smaller than 1 mm must be clearly visible in the image.

Throughout the thesis images from this camera system are used. Figure 3.1 displays an

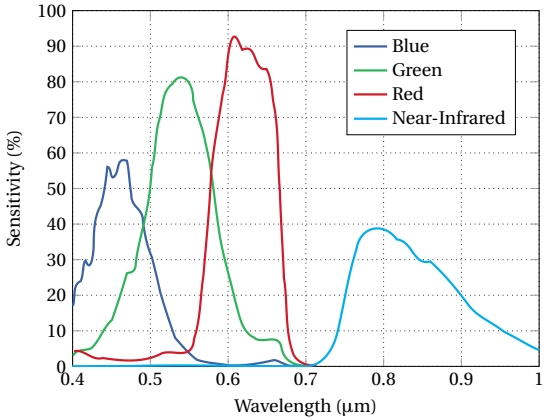


Figure 3.4: Spectral properties of Jai camera: The sensitivity of the CCD sensors in the different bands is plotted. The sensitivity in the NIR spectrum is lower than in the visible wavelengths. Own plot based on data from [136].

example image (left) in a field scenario which was acquired with this camera. In order to reduce the effect of the environment (wind, direct sunlight, etc.) the area around the camera system is shaded and halogen light is installed.

In general the methods developed are not restricted to images from this exact setup. The methods work with any multispectral image with color and near-infrared channel of suitable quality. This includes for example that no displacement between color and near-infrared channel should be present.

3.2 Vegetation Segmentation

In many agricultural image processing tasks only the plants are of relevance and thus a segmentation between the vegetation pixels and the background is desired. The process of performing such a discrimination between vegetation and background is thus called *vegetation segmentation*.

The output of vegetation segmentation is a pixel mask, which masks all pixels that do not belong to vegetation. Sometimes the opposite result is desired: When the mask is inverted, only background or soil pixels are retained in the image. For example when soil properties are of interest and soil pixels are analyzed with computer vision, vegetation segmentation is also performed for the inverse result.

Vegetation segmented images can be used for different purposes besides computer vision, for example for visualization and manual visual inspection. In this thesis vegetation

segmented images are the input for the plant classification and detection processes that are introduced in Chapter 4 and Chapter 5.

In the remainder of this section related work and the developed method for vegetation segmentation are discussed.

3.2.1 Related Work

Research on vegetation segmentation is an established, but active domain: Due to recent improvements in camera technology, different field or environment conditions and different requirements regarding the desired output, novel vegetation segmentation approaches are continuously presented. However, in most publications vegetation segmentation is considered a preprocessing step and is only briefly discussed. The vegetation segmentation approaches thus are often presented for special use cases and for special camera equipment.

In the following vegetation segmentation methods are reviewed and grouped according to the applied input data (for example color images vs. hyperspectral data) and the segmentation method which is applied (for example color-based thresholding vs. classification-based methods).

A first group of approaches relies on color images from cameras or video image streams. The vegetation segmentation is then performed using color indices and thresholding [137, 26, 23, 138, 139, 140, 141] or clustering-based on a color model [140]. Here the focus lies on methods using only RGB images — in the upcoming Section 3.2.2 the extension to vegetation indices for multispectral images is introduced together with the well-known Normalized Difference Vegetation Index (NDVI).

Woebbecke et al. introduce the Excess Green Index (ExG) approach for vegetation segmentation [137]. The approach works for color images by computing the ExG index for each pixel according to the formula:

$$\mathrm{ExG} = 2G - R - B \tag{3.1}$$

The segmentation in vegetation and background is then performed by thresholding of the ExG values. Pixel with intensity values below the threshold are background, pixel with intensities above the threshold are vegetation.

Many variants of the ExG index by Woebbeke et al. were proposed, for example the Excess Red Index (ExR), which is defined as follows:

$$\mathrm{ExR} = 2R - G - B \tag{3.2}$$

Neto et al. present a leaf segmentation approach where in a preprocessing step background pixels in the images are removed through a combination of the ExG and ExR

indices [23]. The difference image ExG − ExR is processed with connected component analysis (to cope with multiple leaves or leaf fragments) and then further processed with their additional leaflet segmentation approach (which is beyond vegetation segmentation).

Perez et al. apply the Normalized Difference Index (NDI) [26] to segment vegetation from background.

$$\mathrm{NDI} = \frac{G - R}{G + R} \tag{3.3}$$

Meyer & Neto compare the already presented vegetation indices and propose a zero threshold version of ExG − ExR [138]. Zero threshold means selecting 0 as threshold parameter (independent of the image data). However, for the other methods they compare against they apply a data driven method to select the threshold; i.e. in the end all three methods do not require a hand tuned threshold. In their experiments the ExG − ExR method performs slightly better than NDI or other indices.

The approach by Bai et al. applies k-means clustering to detect vegetation in color images [140]. The RGB images are converted to LAB color space before processing. Their approach features an offline stage for color model and parameter estimation (given hand labeled images) and an online segmentation stage where new images are processed. For images with rice and cotton plants they report results of 88.1 % and 91.7 % segmentation accuracy.

Additionally, RGB thresholding-based methods are presented for a variety of special use cases: for example leaf extraction [139] and with focus on shadow resistance [141].

A group of approaches achieves the segmentation between vegetation and background in color images with classification-based methods [142, 143, 144, 145, 146, 147].

The work by Zheng et al. discriminates vegetation through mean-shift and neural network classifiers applied to color features extracted from the RGB and HSI channels [142] . For a 100 image sample dataset acquired in field conditions they report a mean misclassification error of 4.2 %. In the follow up paper [143] mean shift segmentation is combined with the fisher linear discriminant which improves the segmentation accuracy compared to the earlier paper, especially in regions with shadows. In both papers the authors point out that the computation time for their approach is significantly too long for real time applications.

Guo et al. present a method for vegetation segmentation in RGB images which is based on decision trees and additional noise reduction filters [144]. They split the images into blocks of 5×5 pixels and extract 18 color features. In experiments with wheat images, segmentation accuracies of 75.6 % to 87.0 % are achieved and ExG or ExG − ExR methods are outperformed. They conclude that the error rate is still too high for practical application and further improvement is necessary.

Keller et al. compare different machine learned methods with a HSV thresholding based approach for vegetation segmentation and soybean leaf area coverage estimation [145].

In their experiments the HSV thresholding approach outperforms the Random Forest and neural network based approaches.

Additionally, other machine learning methods were used for vegetation segmentation: Campos et al. introduce bag of words classification [146] and Romeo et al. apply fuzzy logic [147] for vegetation segmentation.

Besides these major approaches several other methods were proposed for vegetation segmentation:

Suzuki et al. use a hyperspectral line camera and the NDVI approach (ratio between read and near-infrared, see Section 3.2.2 below) to segment vegetation from background [148]. The threshold was determined manually and kept fixed. The drawback of such camera systems is that they only capture a slit image (one line) and must be moved to get a full field image. Thus non static scenes and shaking motion of the camera introduce distortion into the image.

Marchant et al. apply a special image sensor which delivers red, green and near-infrared information [149]. Then they use a ratio of near-infrared and green to segment vegetation from soil.

Additionally, Nguyen et al. apply an active lighting 3D camera system (color camera combined with active 3D time of flight camera by PMD) for vegetation segmentation of images with bushes and grass [150]. Their approach utilizes the 3D structure of the leaves and branches in addition to color information. Results indicate a true positive rate of 91 % for outdoor experiments with a small field robot. It is however not discussed whether the approach works for downward looking images in crop fields.

In summary many vegetation segmentation approaches were developed. Most utilize color cameras and a combination of color information and either thresholding or classification-based methods. The majority of approaches is tailored towards a specific field application. In the following a vegetation segmentation method is developed, which makes use of a multispectral camera system.

3.2.2 Normalized Difference Vegetation Index

Vegetation indices were developed within the remote sensing community for vegetation detection with spectrometers from air or space borne systems [151]. The basic approach of a vegetation index is to exploit the distinctive reflectance properties of vegetation in different bands (wavelength) for the discrimination from soil and other objects such as stones. A multitude of vegetation indices exist for a variety of sensing systems on the one hand and different purposes and applications on the other hand (for example vegetation detection or water and nutrient content estimation in leaves).

A well-known and widely used vegetation index for multispectral images with color and near-infrared information is the *Normalized Difference Vegetation Index* (NDVI) [152]. The

NDVI is calculated from the reflectance in the red (R_{R}) and near-infrared (R_{NIR}) band according to the NDVI formula (Equation (3.4)).

$$V_{\mathrm{NDVI}} = \frac{R_{\mathrm{NIR}} - R_{\mathrm{R}}}{R_{\mathrm{NIR}} + R_{\mathrm{R}}} \tag{3.4}$$

The specific wavelengths at which the reflectances are measured differ, sometimes narrow-band measurements are taken very close to the red edge, other approaches use measurements further from the red edge with wider bands.

Equation (3.4) can only be used directly when the sensor measures reflectance values of the objects it perceives. However, cameras do not measure reflectance, but deliver an intensity value. This intensity measurement depends not only on the reflectance of the object, but also on the ambient illumination, the camera's optical path and image sensor.

Therefore the NDVI formula is adapted to images: An NDVI image I_{NDVI} is calculated from the input channels I_{NIR} and I_{R}.

$$I_{\mathrm{NDVI}} = M\left(\frac{I_{\mathrm{NIR}} - I_{\mathrm{R}}}{I_{\mathrm{NIR}} + I_{\mathrm{R}}}\right) \tag{3.5}$$

The operators $-$ and $+$ operate on each pixel of the images separately. The function $M()$ maps the NDVI values, which are in the range $[-1;1]$, back to monochrome image values in range $[0,1]$. The assumption is that the input monochrome images (I_{NIR} and I_{R}) are also expressed as floating point images in range $[0,1]$.

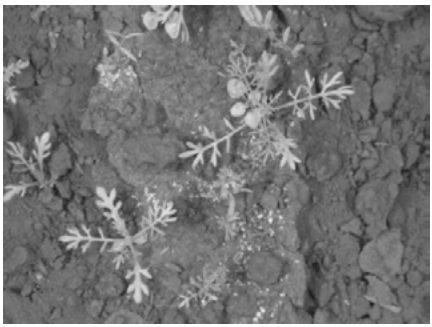
To improve the NDVI image for images from the introduced JAI multispectral camera, the systematic difference in intensity between the red and near-infrared channel can be compensated: As depicted in Figure 3.4 the sensitivity in the near-infrared channel is lower than the sensitivity in the red channel. Additionally, the optical system (lens and additional filters) as well as a difference in illumination in the red and the near-infrared band create this intensity difference.

The NDVI formula in Equation (3.5) is modified and a parameter κ which scales the intensities in the near-infrared image is proposed. The parameter κ can be hand tuned or estimated from data. More details on how to chose κ is given below.

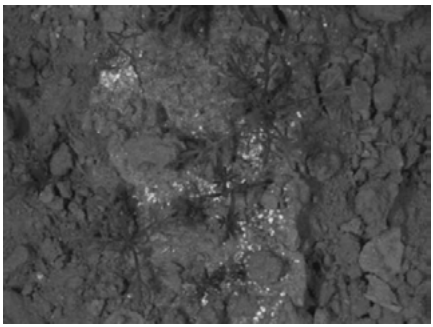
$$I_{\mathrm{NDVI}} = M\left(\frac{\kappa \cdot I_{\mathrm{NIR}} - I_{\mathrm{R}}}{\kappa \cdot I_{\mathrm{NIR}} + I_{\mathrm{R}}}\right) \tag{3.6}$$

Figure 3.5 displays a sample red and near-infrared image together with the NDVI image derived according to Equation (3.6). For better readability the monochrome NDVI image (c) is also printed in the jet color map (small values blue, large values red).

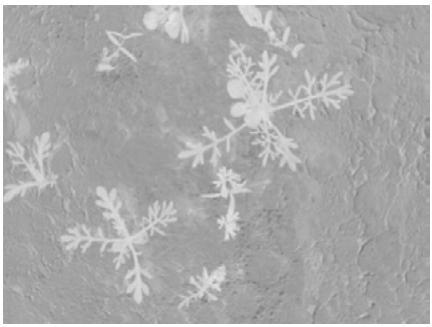
In the following, the parameter κ is derived from an image showing only soil. The value for κ is chosen such that the NDVI values of all soil pixels average to 0.



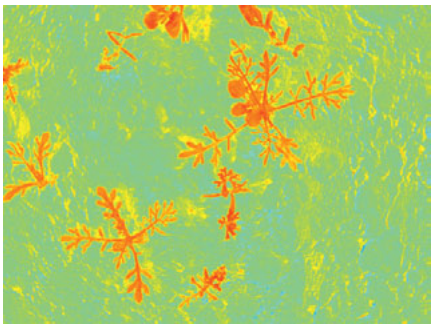
(a) NIR channel



(b) R channel



(c) NDVI image



(d) NDVI image in jet colormap

Figure 3.5: Near-infrared image (a), red image (b) and composed NDVI image in monochrome (c) and jet (d) color map.

3.2.3 Segmentation and Filtering

The separation of vegetation and soil is achieved by thresholding the NDVI image to calculate a soil mask.

Figure 3.6 displays the NDVI values of the example image Figure 3.5 (c). Two peaks are visible in the histogram: the high peak at an NDVI value of approximately −0.1 stems from the background pixels and the lower peak at an NDVI value of approximately 0.4 stems from the biomass pixels.

Threshold-based Segmentation The NDVI image is well suited for this background masking operation using a threshold because soil pixels have lower NDVI values than vegetation pixels. This soil mask is subsequently applied to the NDVI image to mask pixels that belong to the background and do not display plants.

The threshold which is used to generate the background mask is calculated using Otsu's

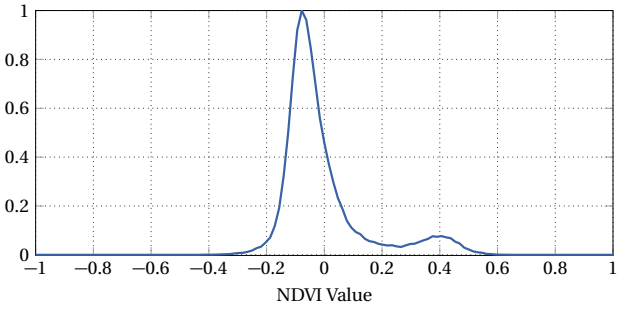


Figure 3.6: Histogram plot of NDVI values from example image Figure 3.5 (c). The NDVI values expose a bimodal distribution. It is composed of a large peak (background soil pixels) and a smaller peak (biomass).

method [79]. This method calculates an optimal threshold under the assumption that the input data is a bimodal distribution.

The threshold is selected to minimize the within class variance σ^2_{w}. Otsu shows that minimizing the within class variance is equal to maximizing the between class variance σ^2_{b} which can be calculated more easily:

$$\sigma^2_{\mathrm{b}} = w_0(t)\, w_1(t) \left[\mu_0(t) - \mu_1(t)\right]^2 \tag{3.7}$$

First, a normalized histogram of the data is calculated where p_i is the normalized count of pixels with an intensity which falls into bin i of the histogram and L is the number of bins. Without loss of generality, a bin is created for each of the 256 intensity values in an 8 Bit image. Now the weights $w(t)$

$$w_0(t) = \sum_{i=1}^{\mathrm{t}} p_i \tag{3.8}$$

$$w_1(t) = \sum_{t+1}^{\mathrm{L}} p_i \tag{3.9}$$

and class means $\mu(t)$

$$\mu_0(t) = \sum_{i=1}^{\mathrm{t}} \frac{i\, p_i}{w_0(t)} \tag{3.10}$$

$$\mu_1(t) = \sum_{t+1}^{\mathrm{L}} \frac{i\, p_i}{w_1(t)} \tag{3.11}$$

can be computed for a selected threshold t and the p_i values from the histogram. Then the optimal threshold t_{otsu} is selected:

$$t_{\text{otsu}} = \arg\max_{t} \sigma_{\text{b}}^2(t) \tag{3.12}$$

Mask Generation The mask I_{mask} is derived by applying the threshold t_{otsu} to all pixels p_{NDVI} in the image I_{NDVI} according to Equation (3.13).

$$p_{\text{mask}} = \begin{cases} 1, & \text{if } p_{\text{NDVI}} > t_{\text{otsu}} \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

An improvement can be achieved along the borders of plants when the NDVI image I_{NDVI} is smoothed slightly with a Gaussian blur kernel before the thresholding Equation (3.13) is applied.

Mask Improvement To further improve the background mask, small blobs are removed from the mask. For all blobs in the mask the area of the blob is calculated; if the area is below a threshold t_{area} the blob is deleted from the mask.

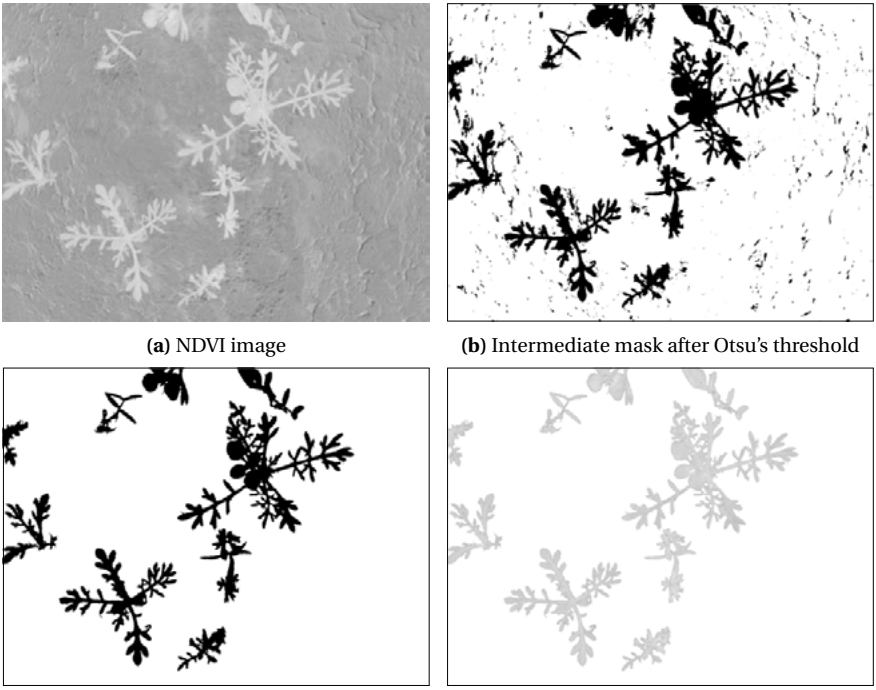
Such a filtering approach helps to suppress blobs which are too small to be considered a plant or leaf fragment. The minimum size of such fragments can be easily set a priori by defining the minimum size of objects in mm^2; then this is converted into pixels given the camera setup properties and the projective formulas. Example cases for which this filtering step is helpful are small stones or objects, specular reflections and tiny vegetation or wood fragments.

Additionally, image regions where the intensity in the NIR image is below a threshold t_{nir} are added to the mask such that they are removed from the final image. This improves the mask especially in areas at plant centers where shadows change the intensity value of the red channel.

The output of the background removal step is a background soil mask image I_{mask}. This mask is applied to the NDVI image and the resulting masked NDVI image $I_{\text{NDVI masked}}$ can be used for further processing with machine vision.

Figure 3.7 displays the example NDVI image and masked NDVI image side by side. Additionally, the mask after Otsu's thresholding (b) is displayed together with the final mask (c). It can be seen that the mask improvement process removes the small blobs present in the intermediate mask as well as a few pixels at plant centers where shadows are present.

Parameter Selection The parameters introduced in this section depend on the camera system being used. For the Jai camera which is used throughout the thesis the following parameterization is chosen: The smoothing kernel width is set to 3, the blob size parameter t_{area} is set to 300 px, the NIR pixel threshold t_{nir} to 25 %.



(a) NDVI image

(b) Intermediate mask after Otsu's threshold

(c) Final mask after improvement

(d) Masked NDVI image

Figure 3.7: NDVI image, intermediate mask, improved mask and final masked NDVI image where all soil pixels are masked with white color.

Throughout the experimentation the threshold t_{otsu} is not calculated for each image separately, rather a single pre-determined threshold is used for a complete dataset. This improves the segmentation for example for images with no vegetation.

This fixed threshold for a dataset or recording session is determined once using one or more images where a substantial amount of vegetation is present (for example the image in Figure 3.7 is well suited) to not violate the bimodal assumption of Otsu's method.

3.3 Summary

This chapter introduces and derives a field image acquisition setup together with a suitable high-performance vegetation segmentation method. In summary the following is achieved:

- A camera setup comprising a multispectral camera and a suitable lens is derived to deliver field images with four color channels (red, green, blue and near-infrared).

  The camera system is able to deliver the multispectral images in one shot at frame rates of 1 to 30 Hz which allows capturing the field images with a continuously moving robot platform.

- The developed vegetation segmentation method based on the NDVI (Normalized Difference Vegetation Index) is able to derive a vegetation mask from the multispectral images.

  The effectiveness of this method is proven by visual inspection and by the high performance results obtained in the depended tasks (see results of Chapters 4 and 5).

- An image dataset with multispectral images and vegetation segmentation masks is published in conjunction with [70].

The resulting field images are multispectral images with a vegetation segmentation mask. The segmented images are an ideal data format for field image analysis, for example data visualization, manual inspection or processing with automatic image analysis algorithms. Also within this thesis this data format is the common input data for the plant classification and position estimation tasks, which are developed in the next chapters.

4 Plant Classification

This chapter introduces a novel plant classification system for plant discrimination in field images. The objective is to process field images which display different plant species and to classify the plants for example into crop and different weed species (see for example Figure 4.1). The desired output is a plant classification image where the estimated plant class is available for every vegetation pixel in the image and the different classes are distinguished through color coding.

Figure 4.1: Input field image (left) and resulting plant classification (right). The plant class is color coded where green denotes the crop class while red and blue denote different weed classes. At the border the system does not output a classification, thus it is plotted in darker gray.

The plant classification approach must be able to cope with the specific situation of outdoor crop/vegetable farming: In general, crops are grown in one or multiple rows, weeds however will occur close-to-crop and both between crop rows (inter-row weeds) and inside the crop row (intra-row weeds), see Figure 4.2 for an illustration of the situations. Additionally, overlap between plants occurs, including inter- and intra-class overlap. Figure 4.1 displays all of these situations: For example in the bottom row, a carrot plant (green) grows between three weed plants (red and blue) and all plants overlap.

In contrast to related work a new approach is developed that enables plant classification in such challenging situations with overlap and close-to-crop weeds: Prior segmentation into individual plants or leaves is not required. Instead, crop and weed are discriminated based on features that are extracted from image patches. The patches are generated from the image using a sparse sliding window approach. Neighboring image patches

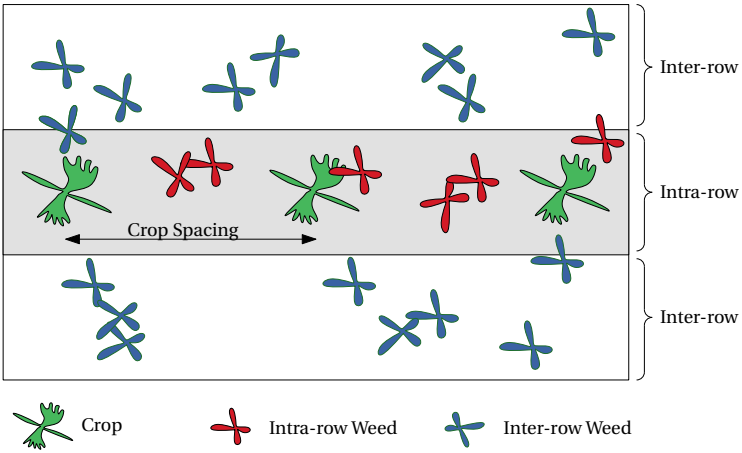


Figure 4.2: Plants are mostly cultivated in rows. In such row-based field setups weeds can be classified to be located inter-row or intra-row. The figure depicts one crop row (gray background); in a field setup the next crop rows would be directly located above and below the depicted inter-row zones (white background). Additionally, overlap between weed/weed and crop/weed is present.

overlap because the patch size is significantly larger than the spacing of the grid. For classification, a supervised machine learning algorithm (Random Forest) is applied and estimates crop and weed scores for each patch location. Subsequent smoothing using a Conditional Random Field and nearest neighbor interpolation yield the final full scale plant classification image.

The plant classification system comprises several processing stages: The major processing steps run online during classification of new images. Two additional offline steps are developed to generate annotated training data and perform classifier training.

In the following sections, first related work is discussed, then all processing steps are described in detail. Intermediate results for every processing step are shown and the parameter selection is done. The evaluation of results with multiple datasets and discussion is conducted in Chapter 6.

4.1 Related Work

Plant and leaf classification with computer vision techniques has been studied before on different levels. Figure 4.3 provides a graphical overview of the four main approaches: Leaf-based classification, plant-based classification, row-based methods and cell-based methods.

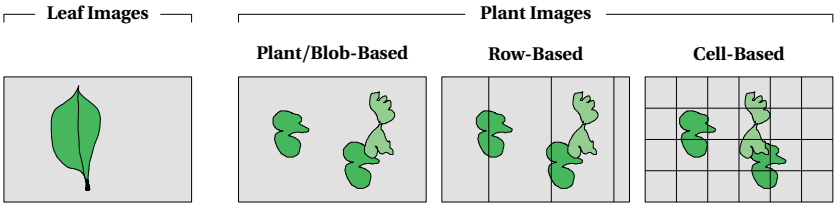


Figure 4.3: Visual illustration of plant/leaf classification approaches in related work: Single leaf in image, segmented blobs, crop row detection and cell-based methods. The crop plant is displayed in light green in the plant images. All four approaches are reviewed in detail in the following sections.

4.1.1 Classification of Leaves in Images

A lot of methods were developed for the classification of leaves with machine vision. Most work focuses on the use case where a single leaf is captured under controlled image acquisition conditions: For example a flat leaf is placed on a uniform background in front of a camera. Most methods apply properties like color, shape and texture to classify leaves from different species [16, 17, 18, 19, 20].

Du et al. classify leaves based on morphological features [16] [17]. The features include shape features (aspect ratio, area of convex hull, circularity, etc.) as well as the invariant Hu moments [153]. On their 20 species dataset they achieve classification accuracies between 68 % and 91 % using a modified hypersphere classifier or nearest neighbor classification.

Liu et al. propose a leaf classification method which uses a wavelet method [18]. The used features for classification are statistics which are calculated from the wavelet coefficients. They apply a kernel SVM for classification and report 95 % recognition accuracy on their 15 species dataset.

Beghin et al. present a method for classification based on the contour signature of the leaf and texture [19]. The contour signatures (distance to the leaf center for N clockwise contour pixels) and the texture histograms (histogram of oriented Sobel gradients) are compared using Jeffry divergence and an incremental classification approach is applied. The system is evaluated using leaf data of 18 species and achieves a classification performance of 81.1 % on their dataset.

Mouine et al. propose an automatic approach for leaf species identification [20]. The classification is based on the leaf contour (shape context feature [154]) and the arrangement leaf salient points (harris corners). Their system obtains classification scores of 58 % and 61 % on the ImageCLEF 2012 [155] plant identification task.

Recently, these technologies were integrated into mobile devices like smartphones [21, 22].

This enables classification of leaves in the field and requires methods that can run on mobile devices with consumer grade camera and computation hardware.

Kumar et al. present a smartphone application which can be used to classify leaves of trees from the Northeastern United States [21]. The app for iOS mobile devices uses features that represent the curvature of the leaf contour in multiple scales to discriminate leaves from 184 different trees. On the test data they are able to identify the leaf correctly within the top 5 matches with 96.8 %. They do not report results for the direct identification result where a single class is estimated for the query leaf.

The ApLeaf application by Zhau et al. is an Android app for leaf identification [22]. After capturing an image, the leaf is segmented from the background, the stem is removed using tophat filtering and features are extracted. The app returns the top matches from a database of leaf images from the French Mediterranean Area (ImageCLEF 2012). They use color statistics, wavelets and pyramid HOG features and evaluate the performance with ROC curves.

Only few work focuses on field-based leaf extraction and classification.

Komi et al. simulate a field setting by placing multiple leaves on a tray with soil. However, the leaves are mostly flat and do not overlap [24]. They use the same shape and texture descriptors as [27] and added eccentricity and a hyperspectral reflectance value vector. They calculate these features for all blobs in the image and classify them into 6 leaf species using linear discriminant analysis (LDA). They report 90.3 % identification rate for their non-overlapping full leaves test data. The effects of challenging field situations like overlap and images of non-flat leaves are not addressed.

Neto et al. propose a leaf extraction method to segment individual leaves from plant canopies in field images with complicated backgrounds [23]. The approach uses color and connected component analysis with fuzzy clustering and a genetic algorithm to separate leaves. The approach is successful for convex leaves; however the authors conclude that more research is required for concave and pinnate leaves (like these of carrot and chamomile plants). Their color and connected component features are not discriminative for these complex leaf shapes and they suggest evaluating shape and texture features.

Slaugther et al. utilize a spectrometer with a spot size of 3 mm to classify the leaf at which the spot is pointed [25]. 21 wavebands are used as features and stepwise discriminant analysis is applied for plant distinction. Experiments are performed with lettuce plants in California and they report an overall crop/weed classification accuracy of 90 %. The spot size of a few mm renders this approach useful only for select measurements. Systematic classification of all plants in a field scenario is unrealistic with a 3 mm spot size. The authors do not mention any extension towards hyperspectral cameras which could mitigate this drawback at the substantial cost of requiring an expensive hyperspectral camera.

These field-based studies all address special scenarios like exposed leaves on known background, only support convex leaf shapes or require special spectrometer equipment.

These methods are not suited for large scale classification of leaves for weed treatment in commercial field scenarios.

4.1.2 Classification of Plants in Images

A second set of methods focuses on classification of plants and not just single leaves. For field-based plant classification different sensing setups are possible. First, airborne or spaceborne sensing systems can be applied for remote sensing [40]. Second, close-range sensing can be applied with ground-based machinery like tractors or robots.

The motivation for high precision plant classification on a per plant basis is site-specific treatment or single plant weed control [2] with ground-based vehicles. To be able to distinguish single plants, a high spatial resolution is required. Therefore, the following review of related work focuses on methods with ground-based sensing which can deliver such high resolution plant classification results.

Ground-based plant classification with camera-like sensors can be divided into three major approaches: Plant/blob-based methods, row-based methods and cell-based methods. These will now be reviewed in detail:

Plant/Blob-based Methods Several methods process the field image into segmented blobs or plants and then subsequently classify each blob/plant based on features that are extracted from each blob [26, 27, 28, 29, 30]. Good segmentation into single plant blobs is a requirement for good classification results; especially overlap and partial plants are challenging situations.

Hemming & Rath present a weed detection system which is integrated into a robotic weed control system [27]. For each blob, shape and color features are extracted and used for weed discrimination. For their experiments in greenhouses and fields, classification accuracies with a wide span from 51 % to 95 % are achieved by their method. The plant segmentation step is identified as error source which reduces the system performance.

Åstrand & Baerveldt developed a similar robot for plant classification [29]. The system also classifies segmented plants with similar features. They achieve classification accuracies between 86 % to 97 % when applying the system to plants with approximately 5 cm in diameter. However, the plant segmentation step was done manually; for the fully automated system they estimate a degradation of the performance of up to -15 %.

Blasco et al. developed a robotic weed control application with a plant classification approach to detect weeds in lettuce fields [28] which is integrated into a field robot. The paper describes the image acquisition and treatment system which is integrated into a robot in detail. The vision system detects the weed plant blobs based only on thresholding in RGB color space. With large lettuce plants (approximately 10 cm in diameter) they were able to locate 99 % of lettuces and 84 % of weeds correctly (which according to the picture are easy to discriminate by color).

Perez et al. apply color information to segment background from biomass and to extract vegetation blobs [26]. Subsequently shape features (area, major axis, moments and derived values) are applied to the segmented blobs. Then the discrimination is performed by using Bayes rule or k-nearest neighbor classification. On the 32 test images the performance of both approaches is nearly identical with true positive rates of 79 % to 89 %. The dataset is however very small and they conclude that further research is required because of overlap between plants and the large size variation of weeds.

Zheng et al. perform corn vs. weed classification in perspective outdoor RGB images [30]. After background removal using ExG, blobs containing vegetation are extracted and color index features calculated by averaging over all pixels in the blob. A one class SVM approach is implemented and results in high accuracies above 90 %. Such high accuracies are achieved since the field situation is simple: Corn plants are large, weed density is low and plants do not overlap in the presented images.

These plant or blob methods all face problems, when the blob segmentation is flawed, for example in situations with plant overlap or when plants are difficult to segment from the background. Especially in outdoor fields and when vegetables are cultivated such situations occur and are a challenge for all blob-based methods.

Row-based Methods Another approach to plant classification in agriculture are row-based methods [31, 32, 33, 5, 34]. Most crops are cultivated in rows, where crop plants are sown in straight parallel lines: weed plants on the contrary occur everywhere. Row-based methods utilize the row information to classify plants into weeds (growing in between rows) and crops (growing in the detected row).

Onyango & Marchant propose a system based on color and morphology to differentiate between crop (cauliflower) and weed [31] in advanced growth stages. In addition to the color and shape information they make use of the cauliflower planting grid as crop prior information which is modeled as a bivariate Gaussian distribution. For their 12 test images they achieve classification rates of 82 % to 96 %.

Gee et al. estimate crop rows in perspective color images using the Hough transform [32]. Subsequently, blobs are extracted and classified into crop and weed based on the row information. They conclude that the Hough-based method works well when a precise vegetation segmentation is available. The blob-based method only allows detection of inter-row weeds and they propose to evaluate spectral based classification of the blobs to also be able to detect weed growing inside the row. Performance is estimated for soy and corn crops in different scenarios and classification performances of 89 % to 96 % for crops and of 74 % to 91 % for weeds are achieved.

De Rainville et al. develop a crop/weed classification system using morphological analysis of weed in crop rows [33]. Color images are first segmented into a vegetation image, then crop rows are detected using the Hough transform. Subsequently, blobs are identified and described with morphological features (for example area, compactness, mayor axis, connectivity). Then a naive Bayes classifier is combined with a Gaussian mixture model to discriminate crops and weeds. Thus the distribution of weeds inside and

outside the crop row is modeled and used as additional information.

Pena et al. apply unmanned aerial vehicle based sensing for weed mapping [5]. They apply the NDVI segmentation, crop row identification and discriminate weed and crop based on the row structure. The final weed map is output as coarse row aligned cells.

Additionally, publications exist which just focus on row detection (using different methods like Hough lines, vanishing points, frequency analysis and others) and use the row information for steering and robot guidance [53, 156, 54]. More recent methods can also cope with curved rows and model row extraction with fitting polynomials [55].

The disadvantages of row-based methods is that often intra-row weeds are not detected and treated. In high value crops like vegetables there is a high intra-row weed pressure and those weeds must be detected and treated to avoid yield losses.

Cell-based Methods To avoid the plant or row segmentation step in field image scenarios, cell-based methods were developed [35, 36, 38, 37]. The image is split into large non-overlapping cells through tessellation. Then all processing steps and the classification decision are made for each cell (for example whether the cell contains weed or not). The output is restricted to a per cell output; for example a cell can be determined to be weed infested and as a result herbicide will be applied to the cell.

The method by Aitkenhead et al. splits each image into 16 coarse cells (4 rows and 4 columns per image) where each cell covers approximately 4 cm by 3 cm of the growing tray. [35]. The crop/weed decision is derived per cell using self-organizing neural networks. In experiments with specifically sown plants in a greenhouse this approach achieves approximately 75 % classification accuracy.

Tellaeche et al. developed a cell-based machine vision system for post-emergence herbicide application [36]. The field images from a frontal downward looking camera are perspectively corrected and aligned to rows. Then, each cell is classified whether it contains weed or not and a Bayesian classifier determines weather cells are sprayed or not.

In another more recent paper Tellaeche et al. [37] expanded their system and apply a kernel SVM to two weed coverage features extracted from each cell. The weed coverage features are derived from the distribution of vegetation pixels in each cell with the knowledge that each cell spans the area between two crops rows. In the scenario where a special weed (avena sterilis, which looks very similar to crops) has to be detected they report classification accuracies of 66 % to 85 %.

Nejati et al. present a robot-based system which uses cell-based classification to detect small weed in corn fields [38]. They tessellate the image into small cells (approximately 12–15 cells per corn plant) and analyze each cell using Fast Fourier Transform (FFT) and leaf edge density. They report classification accuracies of 92 % for experiments with their 80 image dataset of corn plants. A novel idea of Nejati et al. is the use of smaller cells and a heuristic filtering (they call this a cell-by-cell check) where miss-classifications inside a region that otherwise consistently is classified as for example crop are removed.

One drawback of all cell-based approaches is a reduction of the spatial precision. The decision result is not available on a pixel or plant level, only on a coarse cell level. Such a cell-based output is well suited for herbicide applications where regions are either sprayed or not sprayed. For high precision phenotyping application where individual plant-related results (for example single plant weed control, plant counting, leaf area measurement) are desired cell-based methods are not directly applicable.

Other Methods Also other techniques can be used to classify plants in fields: Multi- and hyperspectral imaging [41, 42, 43] and remote sensing [40]. Furthermore, Strothmann et al. apply a novel multispectral 3D sensing method and perform pixel-wise plant classification with grid aggregation; the output is similar to the cell-based methods discussed above [39].

Finally, also non-computer vision approaches with for example RTK GPS were proposed for plant classification [44, 45, 46]. The basic idea is to create a high precision seed map while sowing and then using the seed map and RTK GPS to relocate the crops during weed treatment. This works only well for larger plants, as the precisions of RTK GPS is typically in the 2/4 cm range under ideal conditions.

Besides the plant classification application these methods can also be modified and applied to other task in agriculture. For example defects on vegetables can be determined with machine vision [47] and flowers can be analyzed with the goal of determining whether single plants have diseases [48]. Also robotic harvesting of small crop such as sugar snap peas [49] is under development and benefits from computer vision for crop classification.

In addition, the plant species is not the only measure that is of interest to farmers [50]. Metrics like the number of plants per square meter, the height of plants, nitrogen or water content, etc. can be derived from images using machine vision [51, 52].

4.1.3 Summary of Related Work

The study of related work shows that plant classification has been studied on different levels, but also indicates that plant classification in outdoor fields with close-to-crop weeds is challenging and an open research question.

Simple leaf-based methods work well for species recognition of single leaves but not in field scenarios. Plant-based methods work well when single plants are visible in images. Blob-based methods as well as row-based methods struggle with plant overlap and close-to-crop weeds. Additionally, cell and plant-based methods have the disadvantage that only for discrete cell or pre-segmented plants a classification result is obtained. A variety of other methods for example rely on complex and expensive hyperspectral sensing, remote sensing or use GPS-based methods which rely on mapping crop plants and therefore cannot produce a full plant classification image.

The new approach presented in this thesis closes the gap of plant classification for challenging outdoor fields with presence of differently sized plants, overlap and close-to-crop

weeds. The developed method avoids segmentation into plants or leaves which was determined a major problem in the literature. Although working without plant/leaf segmentation, the drawbacks of cell or row-based methods are avoided. Through smoothing and interpolation the system is still able to return a consistent per-pixel crop/weed classification result in full input image resolution.

4.2 Novel Plant Classification Pipeline

To solve the task a plant classification pipeline using improved computer vision and machine learning techniques is developed: The novel plant classification pipeline processes input images (vegetation segmented multispectral images) into plant classification images where each pixel of an image is classified into different plant classes.

The plant classification pipeline comprises five online and two offline steps which are displayed graphically in Figure 4.4. The online steps perform the following computations: First, patches are extracted from the images. Then, features are derived from each patch and stored in a database (DB). Third, using a pre-trained supervised classifier for each patch the plant class is estimated. Finally, smoothing and interpolation are the final two online steps which generate a smooth full plant classification image. The goal of the two offline stages is the generation of labeled training data (using a label tool and a human expert) as well as the training of the classifier.

Another important characteristic of a classification system is the number and choice of different classes. A common approach to plant classification is the discrimination into two classes: crop and weed. However, in order to be as general as possible, more than two classes are supported by the plant classification pipeline. For example it is desirable to perform multi-class plant classification and not only binary crop/weed discrimination. This enables phenotyping applications or the detection of specific weeds in the field. The exact choice of plant classes is application specific and the plant classification pipeline can be configured to use two or more classes as needed.

Throughout this chapter without loss of generality three plant classes are considered. The carrot plants which are cultivated on the field are the first plant class called *crop*. The weed plants are split into two classes for the following reason: Chamomile plants have leaves with a similar pinnate contour and shape as the carrot plants. Therefore, chamomile plants form a separate weed class named *chamomile*. Finally, all other weed plants form the *weed* class.

In the following sections the processing steps which are applied to new images (online steps, left part in Figure 4.4) are introduced.

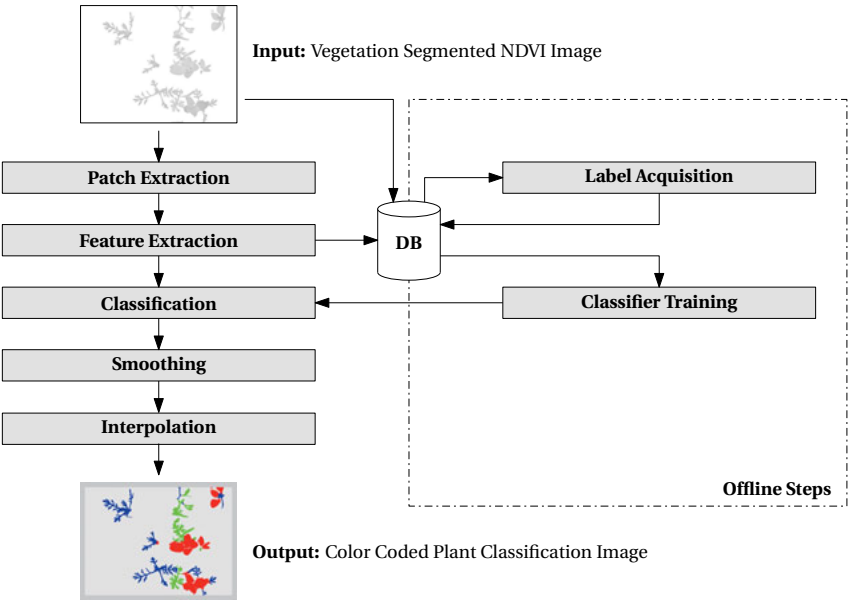


Figure 4.4: Processing steps of the plant classification system. The pipeline processes the segmented NDVI images into a color coded plant classification image. The processing steps comprise an online (left) and offline (right) stage. All processing steps are introduced in detail in the remainder of the chapter.

4.2.1 Patch Extraction

Compared to earlier work where the input image is either segmented into plants or split into cells, a new patch-based approach for plant classification is proposed. The input to this first stage in the plant classification pipeline (see Figure 4.4) is an image with only plant pixels; for example acquired and preprocessed with a system which is presented in Chapter 3. The patch extraction step splits the image into overlapping patches.

The motivation to split the image into patches is to divide and conquer: Instead of processing the complete image which contains multiple, possibly overlapping objects, the image is split into smaller patches. The classification task is done on patch level and later in the processing pipeline (Section 4.2.4) these classification results on patch level will be fused to a full classification result for the whole image.

The patch extraction step is visualized in Figure 4.5 and realized as follows: The individual image patches are extracted from the image using a sliding window approach with additional filtering; in the following this method is called a *sparse sliding window*. The additional filtering step consists of a decision based on the window content whether to

extract the patch or not. For plant classification, the filtering step rejects window positions where no biomass is located at the window center pixel.

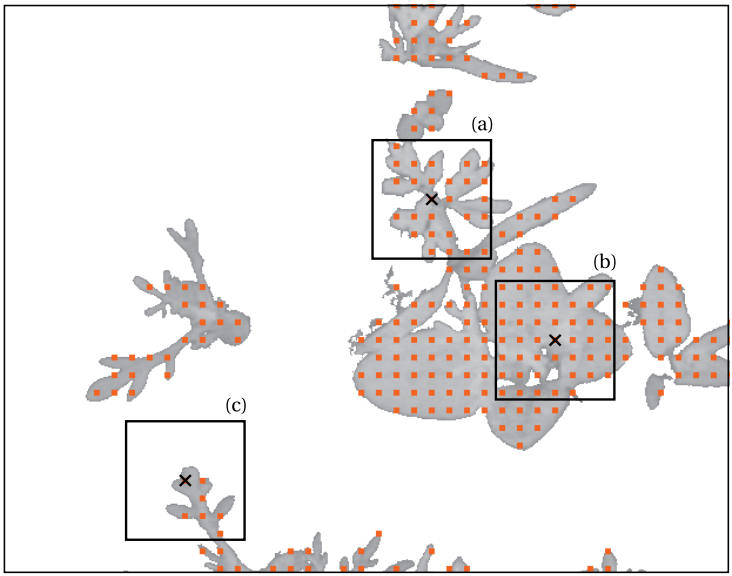


Figure 4.5: Patch extraction is performed using a sliding window scheme. If the patch center (orange dot) displays biomass, a patch is extracted. Three exemplary patches are highlighted and a cross marks the corresponding keypoint for each patch.

In addition to the image patch content the position of the center pixel of the patch in the complete image is saved. This position is expressed in image coordinates (u, v) and called keypoint (see Section 2.1.3 for definition) of the patch. Later, all calculations based on the image patch are related to this keypoint, which describes the patch location in the original input image.

The important parameters of the patch extraction step are

- the size of the patch: w_{size}
- the stride between adjacent window positions: w_{stride}
- and the filtering function to reject window positions without plants.

The patch size must be chosen according to the camera setup and the size of the plants so that a patch contains a part of a plant/leaf. The stride is set to a value smaller than the patch size to ensure that the neighboring patches overlap. Without loss of generality the stride and patch size are equal in horizontal and vertical direction. A simple filtering function is

chosen for the plant classification pipeline which leverages the biomass segmentation: A window is rejected if the center pixel does not contain vegetation.

The output of the patch extraction step is a list of N tuples (P_i, k_i) where P_i denotes a patch and k_i a keypoint. The keypoint k_i is given in image coordinates (u_i, v_i).

$$[(P_i, k_i)] \qquad \text{with } k_i = (u_i, v_i) \text{ and } i = 1, \ldots, N \tag{4.1}$$

Figure 4.6 displays the example patches which are marked in Figure 4.5.

(a) Crop **(b)** Weed **(c)** Chamomile

Figure 4.6: The figure displays the three example patches from Figure 4.5. The colored borders of the patches denote the plant class type which is also given in the sub captions.

The sparse sliding window method is a key contribution: Compared to other approaches on the one hand the error-prone segmentation into plants is avoided and on the other hand compared to cell-based methods the output is much denser (because of the overlap). The experimental results section will show that such a patch-based representation is useful to classify plants in field images.

The following processing steps now operate on the image patches. A final processing step will convert the per patch classification results back to a full classification image where a estimated plant class is output for each biomass pixel (see Section 4.2.4).

4.2.2 Feature Extraction

The goal of the feature extraction step is to find a characteristic representation that is useful to discriminate different plant species also under variation of the plant (size, overlap with other plants, orientation of leaves, etc.) and image acquisition conditions. During feature extraction the content of all image patches P_i is described with a numeric feature vector $\boldsymbol{f}_i$. The feature vector $\boldsymbol{f}$ consists of up to K extracted features $\boldsymbol{f} = f_1, \ldots, f_K$.

The features developed for the plant classification task on the one hand exploit the shape of the (partial) plant/leaf that is contained in the image patch. On the other hand, the intensities in the patch are exploited by deriving several statistics from all patch pixels displaying plants.

Table 4.1 summarizes the 15 specific features that are extracted from each patch. The first seven features f_1 to f_7 are shape and contour features which are extracted from a binary version of the image patch (biomass vs. soil). Similar features have been used

Table 4.1: Features extracted from patches for plant classification.

f_i	Description
f_1	Perimeter (Length of Contour)
f_2	Area (Number of Pixels Covered by Biomass)
f_3	Compactness (Area / Perimeter^2)
f_4	Solidity (Area / Area of Convex Hull)
f_5	Convexity (Perimeter / Perimeter of Convex Hull)
f_6	Length of Skeleton
f_7	Length of Skeleton / Perimeter
f_8	Minimum of Biomass Pixel Intensities
f_9	Maximum of Biomass Pixel Intensities
f_{10}	Range of Biomass Pixel Intensities
f_{11}	Mean of Biomass Pixel Intensities
f_{12}	Median of Biomass Pixel Intensities
f_{13}	Standard Deviation of Biomass Pixel Intensities
f_{14}	Kurtosis of Biomass Pixel Intensities
f_{15}	Skewness of Biomass Pixel Intensities

in previous work; however there they were applied to whole plants or leaves and not to cropped patches [27, 17]. The skeleton-based features f_6 and f_7 are added to the well known shape features for this use case. The next eight features f_8 to f_{15} are statistics (minimum, maximum, mean, standard deviation, skewness, etc.) of the intensities of the plant pixels in the image patch. The motivation for such features lies in the observation that weed and crop plants look dissimilar in the NDVI grayscale image.

The features can also be evaluated on how much they contribute to the plant classification task. The feature importance in Figure 4.7 is generated using a Random Forest classifier and by determining the mean decrease of average accuracy. This mean decrease is calculated on the out-of-bag data for each feature separately when its features values are randomly permuted [157]. It can be concluded that the feature f_3 contributes the most while feature f_{10} is the least relevant. For this feature importance evaluation the complete dataset A is used which will be introduced in Section 6.1.

4.2.3 Classification

The task of the classification step is to discriminate the image patches into the different plant classes. For this task machine learning is used to derive a model from training data that allows automatic separation of the different plant classes in new images (see also Section 2.2.3 where classification and machine learning are introduced).

To be able to represent more than one weed class, a multi-class classifier and a supervised

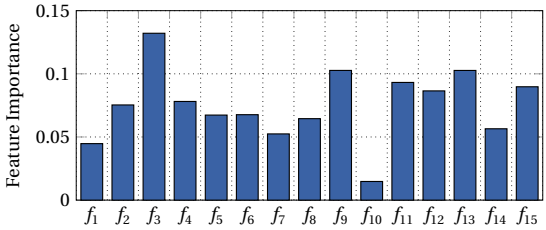


Figure 4.7: Feature importance of the plant classification features for dataset A. The feature importance (y-axis) is the mean decrease in accuracy when f_i is permuted during calculation of the out-of-bag error.

training scheme are selected. In this work the Random Forest classifier (see Section 2.2.5) is chosen, because it supports multiple classes, is fast to train (computation time and regarding number of training samples required) and able to estimate class certainty scores in addition to the most certain label during classification of unseen samples. The training of the Random Forest algorithm for plant classification is one of the offline steps and is explained in Section 4.3.2.

The developed plant classification system is not restricted to using a Random Forest classifier: Any supervised classification algorithm that supports multiple classes and class probability estimates can be applied as a replacement for the Random Forest classifier.

The output of the classification step are a plant class estimate l and a class certainty score vector $\boldsymbol{s}$ (as defined in Section 2.2.1) for each image patch. Figure 4.8 displays the output of the classification step visually: For each image patch the classification result is represented by a colored dot. The color of the dot visualizes the estimated plant class l. In the following, green is chosen for carrot, blue for chamomile and red for all other weeds. The size of the dot indicates the certainty level for this class according to the score vector $\boldsymbol{s}$.

In region A in Figure 4.8, all keypoints are voting for the plant class weed and the classifier has a high certainty. In region B (which displays two crop plants which have overgrown) the classifications are not as certain and several outliers exist. The classifiers estimate varies for different patches which are nearby. This variation in estimated plant class for patches which are nearby is not desired. If the plant classification is taken as input for a weed treatment system, the goal is to have a smooth classification where larger areas are consistent with as few changes of plant type as possible.

During the classification step, all patches are classified independently of each other. This can lead to spatial inconsistencies (different plant classes for neighboring patches) which for example occur in difficult areas where plants overlap or grow close together. The next step will address this and generate a spatially consistent plant classification.

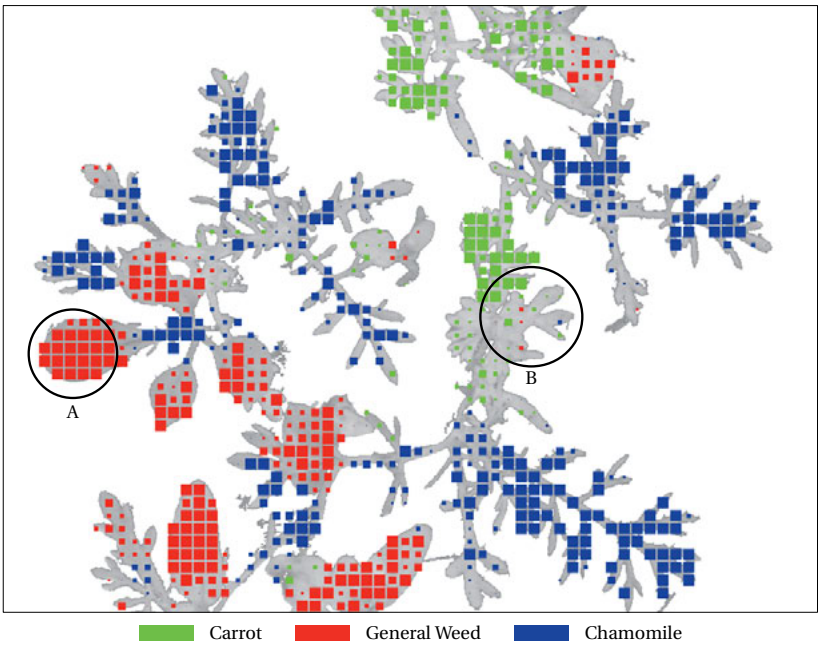


Figure 4.8: Output of the plant classification step: At each keypoint the estimated plant class l is visualized by a color coded dot (see legend). Additionally, the size of the dot indicates the certainty of the estimated plant class according to the score vector $\boldsymbol{s}$. The marked regions A and B are discussed in the text.

4.2.4 Smoothing of Classifications at Keypoints

The classifier provides a plant class score $\boldsymbol{s}$ and most probable class l for every keypoint. Instead of directly taking the most likely class l as estimated plant type for the keypoint the smoothing procedure yields a more consistent plant classification.

Individual keypoints are not completely uncorrelated, rather some keypoint pairs are neighbors in the image. This information is used in the smoothing process: Neighboring keypoints most likely belong to the same class because plants are larger than patches and form continuous objects.

A generic framework which allows the combination of evidence and neighborhood information is the Conditional Random Field (CRF) framework. CRFs are undirected probabilistic graphical models which can take context into account to return the desired more consistent classification [158].

To implement a CRF, the problem is formulated as energy function which is subsequently

minimized during inference. The inputs to the CRF are the estimated score vectors $\boldsymbol{s}_i$ for each keypoint k_i. Additionally, the neighborhood relationship for each keypoint and its eight neighbors is used as a priori information. In the following we call one combination of labels l_i for all keypoints k_i a labeling L. The goal of the smoothing step is to derive the smoothed labeling $\hat{L}$ which is composed of the smoothed labels $\hat{l}_i$ for all keypoints.

The energy function $E(L)$ defined in Equation (4.2) is the CRF model for the plant classification case; it is the weighted combination of a data term D_p and a neighborhood term $V(l_p, l_q)$:

$$E(L) = \sum_{p \in \mathcal{K}} D_p(l_p) + \lambda \cdot \sum_{p,q \in \mathcal{N}} V(l_p, l_q) \tag{4.2}$$

Here, p and q are abbreviations for keypoints. $\mathcal{K}$ is the set of all keypoints k_i in the image. $\mathcal{N}$ defines the neighborhood set; p, q are contained in $\mathcal{N}$ if these keypoints are neighbors. The neighborhood of a keypoint is defined here by the 8 neighbors (up, down and diagonally) of a keypoint.

The data term (Equation 4.3) is based on the estimated class certainty $\boldsymbol{s}_p$ and depends on the currently assumed label l_p at keypoint p. The term $\boldsymbol{s}_p(l_p)$ is the component of the score for the estimated class l_p. Because the score is normalized and sums to 1, the certainty score (higher is better) is transformed into a penalty by subtraction from 1 (lower is better) and forms the data term:

$$D_p(l_p) = 1 - \boldsymbol{s}_p(l_p) \tag{4.3}$$

The neighborhood term (Equation 4.4) is calculated for two neighboring keypoints p and q based on the currently selected labels l_p and l_q at these keypoints:

$$V(l_p, l_q) = \min\left[\left|l_p - l_q\right|, 1\right] \tag{4.4}$$

The discontinuity cost defined by the neighborhood term in Equation (4.4) calculates the integer difference of the integer label classes truncated to at most 1 to only penalize different labels, but not to prefer any class over another. Because the sliding window does not extract patches where no biomass is located at the patch center, a keypoint might have fewer neighbors.

Improved Neighborhood Term The plant classification use case has two special cases which are not considered in the smoothing model so far:

First, background is not modeled as separate class and thus some keypoints for example at plant borders or on pinnate shaped leaves are missing. Especially, for pinnate leaves, keypoints can easily have connectivities which are lower than 8. This might result in less optimal smoothing in these areas.

Second, the neighborhood term does not consider whether the connection between two keypoints (imagine a virtual line between two keypoints) spans only across biomass or whether it includes background pixels.

The first case (border keypoints having fewer neighbors) is improved for pinnate leaves or for plant gaps through an extended neighborhood. The standard 8-connected neighborhood $\mathcal{N}$ is extended to the neighborhood $\mathcal{N}^+$. There missing neighbors are replaced by neighbors in the next row according to Figure 4.9.

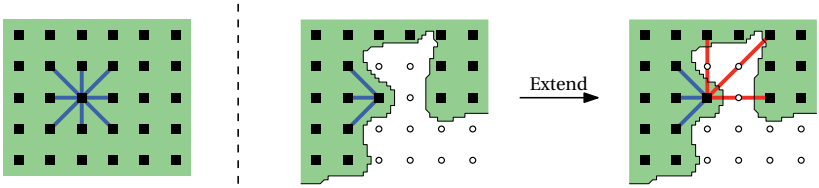
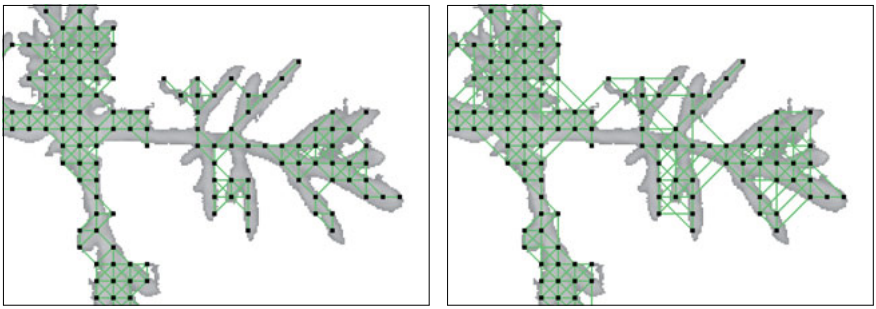


Figure 4.9: Scheme how the neighborhood $\mathcal{N}$ is extended to $\mathcal{N}^+$. This is especially useful to ensure better connectivity at the edges of pinnate leaves, where background is present and keypoints are missing.

The effect of the neighborhood extensions scheme is a better connectivity at the leaf border and in areas where plants or parts of plants overlap. Figure 4.10 gives a real world example where the neighborhood extension is plotted before (Figure 4.10a) and after extension (Figure 4.10b).



(a) Normal neighborhood $\mathcal{N}$.

(b) Extended neighborhood $\mathcal{N}^+$.

Figure 4.10: Real world example showing the neighborhood extension scheme.

The second case which can be improved in the neighborhood term is to introduce a so-called biomass factor. This factor considers the number of background vs. biomass pixels

along the connection between two keypoints (virtual line between two keypoints). The goal is to avoid over-penalizing the neighborhood of two keypoints with different labels when background (non-biomass) pixels are present along the direct connection.

The biomass factor $B_{p,q}$ represents the fraction of biomass pixels along the direct line of pixels between the keypoints p and q:

$$B_{p,q} = \frac{\#\text{plant pixels}}{\#\text{total pixels}} \tag{4.5}$$

Using the biomass factor the neighborhood term in Equation (4.2) is extended: The new discontinuity cost is defined as $B_{p,q} \cdot V(l_p, l_q)$. This new discontinuity cost is no longer binary (either 0 or 1), but now it spans the interval $[0,1]$. If no plant pixels are present it takes the value 0 which results in no penalty if the labels of the associated keypoints are different. If all pixels are vegetation it takes the value 1 and behaves like the initial case defined above in Equation (4.4). For biomass factors between 0 and 1 a linear penalization occurs if different labels are present.

Figure 4.11 displays an example image where the biomass factor is visualized through colored neighborhood connections. It can be observed that connections spanning from one plant to another and also connections along pinnate leaves are weighed down since the biomass factor is lower than 1 for these connections.

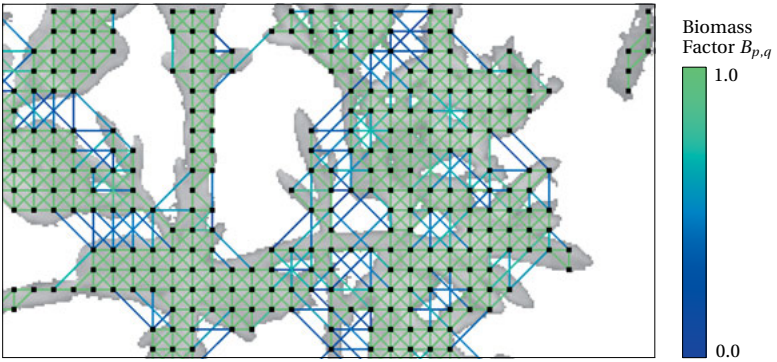


Figure 4.11: Visualization of the biomass factor. Each neighborhood connection is color coded to express the biomass factor value according to the legend on the right.

Final Energy Function Bringing both the new discontinuity cost with biomass factor and the extended neighborhood together yields the final CRF energy term:

$$E^*(L) = \sum_{p \in \mathcal{K}} D_p(l_p) + \lambda \cdot \sum_{p,q \in \mathcal{N}^+} B_{p,q} \cdot V(l_p, l_q) \tag{4.6}$$

The energy function $E^*(L)$ from Equation (4.6) is minimized and the smoothed labeling $\hat{L}$ is determined and returned as result of the smoothing step.

$$\hat{L} = \arg\min_{L} E^*(L) \tag{4.7}$$

Minimizing such energy functions can be done using different algorithms like efficient belief propagation [159] or the graph cut algorithm [160, 161], for which also a sped up variant exists [162]. Multi label graph cuts are used in the following to minimize the energy function. The relevant parameters of the smoothing step are the balancing parameter λ.

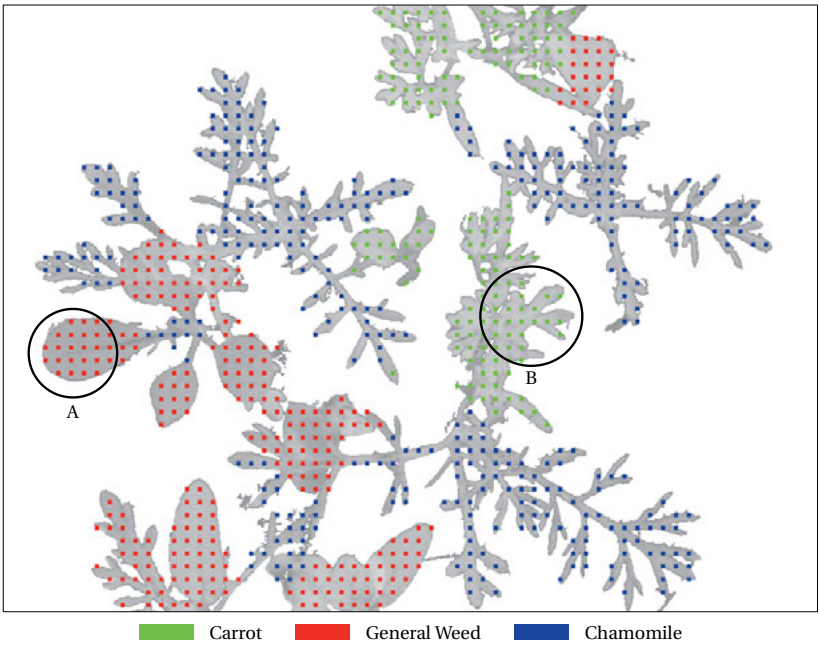


Figure 4.12: The output of the smoothing step for the same image as Figure 4.8: The smoothed version is plotted and the dots are now all of the same size and indicate the result after minimization of the CRF. The marked regions refer back to Figure 4.8 and are explained in the text.

The result of the smoothing process is a consistent labeling. When region B is compared in Figure 4.8 with the results after smoothing in Figure 4.12 it can be seen clearly that the correct crop class has propagated throughout region B. The inconsistencies are no longer present in Region B. Unfortunately, regions which were consistently not correctly classified before smoothing also are misclassified after smoothing (for example the plant

center right next to region A). Region A is already consistent before smoothing and is not changed in Figure 4.12.

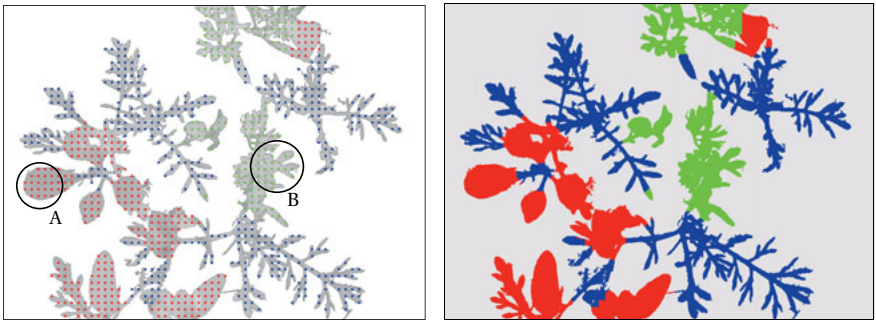
All in all, the output of the smoothing step is a plant classification where for each patch a plant class is estimated. The smoothing step helps to make the individual classifications spatially more consistent and to avoid rapidly changing labels next to each other.

4.2.5 Interpolation of per Keypoint Results to Full Resolution

After the smoothing step, the estimated plant class is available for each keypoint/patch. Compared to the input image, this estimation is still only available for a fraction of the plant pixels. The goal of this step is to yield an estimated plant class for every plant pixel in the image.

The deduction of a full frame plant classification image is implemented as nearest neighbor interpolation. The plant class for each vegetation pixel is copied from the nearest keypoint. Background pixels are not considered in the interpolation step. Due to the patch extraction process no patches are available at the image borders (only full patches are extracted). The final plant classification image has no output at these border pixels.

Figure 4.13 displays the result of the interpolation step for the sample image (from Figure 4.12). Using the nearest neighbor interpolation a full plant classification image is achieved, where a plant classification is available for each vegetation pixel.



(a) Smoothed classification **(b)** Interpolation result

Figure 4.13: Interpolation to full image: The image (a) on the left is the smoothed image from Figure 4.12. On the right in (b) the result of the nearest neighbor interpolation on vegetation pixels can be seen. For color legend see Figure 4.12.

The interpolation step would not be necessary, when a patch would be extracted for each single biomass pixel. This would result in a very large number of patches and calculations which would render the plant classification pipeline very slow. To avoid this while retaining

a plant classification output for each vegetation pixel the patches are sampled from a sparse grid and then the interpolation brings the output back to full image resolution.

4.2.6 Summary Online Steps

The plant classification pipeline comprises 5 major online steps: Patch extraction, feature extraction, classification, smoothing and interpolation to plant classification image. The abstraction from a full image to overlapping patches, their classification and the final conversion back to a full image are the key idea of the system.

A full plant classification image is the output of the five online stages. Figure 4.13b displays an example output image after complete processing with the new plant classification pipeline.

Based on this plant classification image, plant class specific robotic weeding can be realized. In addition, the plant classification image can be used as input for additional phenotyping steps [50]: Metrics like plant count, crop/weed coverage, weed infestation ratio and others can be derived.

4.3 Offline Pipeline Training Steps

The online plant classification process is accompanied by several offline steps. The offline pipeline includes acquisition of ground truth plant classification data and training of the classifier. These steps reuse several building blocks of the online pipeline as depicted in Figure 4.4.

4.3.1 Ground Truth Data Acquisition

The ground truth data for plant classification is generated manually. The field images are presented to one or more experts and each user is asked to mark and classify the plants in the image. Figure 4.14 displays a screenshot of the web-based labeling tool which is based on the open source LabelMe tool [163]. The images that are displayed to the user are the masked NDVI images.

The user then labels the image by drawing polygons and by assigning a single plant class to the polygon. The label of the polygon is assigned to all plant pixels that are enclosed by the polygon. Background non-vegetation pixels that are contained in the polygon (white in Figure 4.14) are not assigned any class by the labeling process and remain background pixels.

During labeling also the number of plant classes is defined. The user can define as many plant classes as he wishes during labeling. When the pipeline reads the labeled data, the number of classes is propagated and set for all further processing steps implicitly.

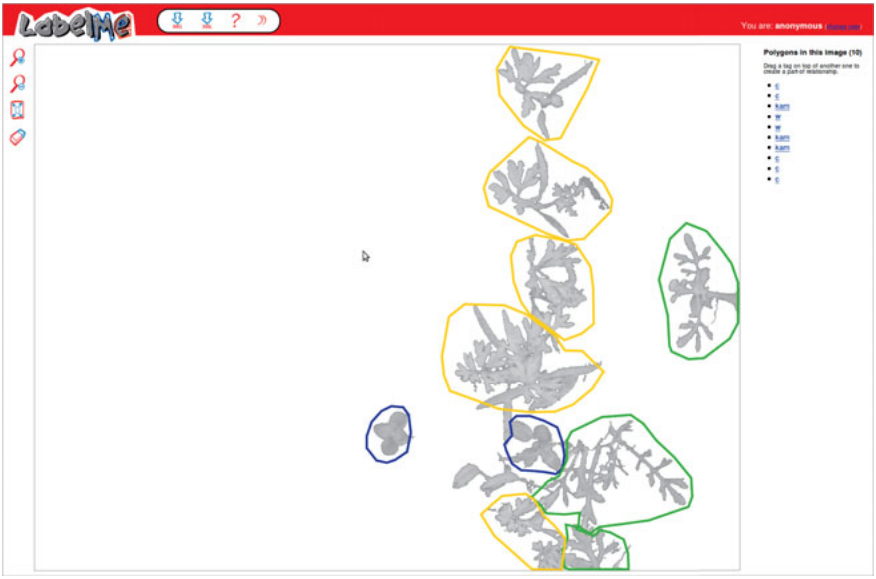


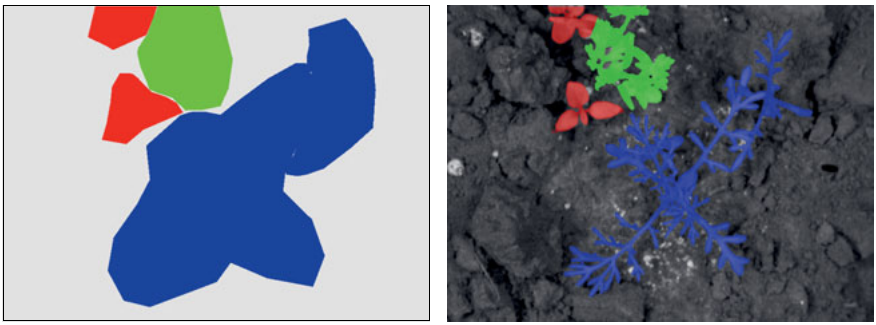
Figure 4.14: The customized web interface used for labeling: The user annotates the images with polygons to define plant contours and sets the plant class per polygon.

As depicted in Figure 4.14 not all pixels in the image must be labeled. For example regions with heavy overlap or at the boundary of the image might be difficult to be labeled correctly and can be ignored at the user's choice. Plant pixels that are enclosed by multiple polygons with different classes are reset to the unlabeled class; this situation can occur when two plants grow together and the labeling polygons intersect.

Figure 4.15 displays the results of the labeling process: Once the user has marked the plant with polygons in the labeling tool, the polygons are combined with the vegetation mask and projected onto the NIR image for visualization and verification.

Here it also becomes clear, that the proposed labeling process is a great support for the expert user. By automatically removing background pixels from the images before labeling is started, the user must not do this manually. Instead of labeling the exact plant contour or assigning the label to all plant pixels with a brush like tool, the user can coarsely label the plant with a rough polygon. Background pixels are automatically ignored. The user is guided to pay most attention to regions with plants growing close together where precise labeling is required.

All labeled images are then stored (with the polygon labels and vegetation mask) in the plant classification database. The database can then be queried for labeled data for the various training, verification or visualization purposes.



(a) Polygon labels from labeling tool **(b)** Labels and mask applied to NIR image

Figure 4.15: Resulting labels for an example image. The polygons which were defined by the user are displayed on the left. When the label polygons are combined with the vegetation mask and are subsequently projected onto the NIR image the labels can be easily verified (right).

4.3.2 Classifier Training

The second offline step is the training of the classification algorithm, which performs the distinction between the different plant classes in the online part of the pipeline (Section 4.2.3).

Using the labeled data a Random Forest classifier is trained in supervised mode. The training process is a two step process, where first labeled data is created from the database and in the second step the Random Forest is trained.

First, all images for which ground truth labels are available in the database are run through the patch and feature extraction process (as presented in Section 4.2.1 and 4.2.2). The resulting feature vectors $\boldsymbol{f}$ for all patches are paired with the ground truth labels g as follows: The keypoint of each patch is used to look up the ground truth label in the database. All feature vectors for which no ground truth exists are discarded at this stage. For quicker retrieval in possible future classification runs, $[\boldsymbol{f}]$ and $[g]$ are cached in the database.

Second, during the training step the database is queried for all labeled training data (feature vectors $\boldsymbol{f}$ with paired ground truth labels g). Using these feature vector and label pairs a Random Forest classifier is trained. The resulting classifier is stored and can then be applied in the classification pipeline.

Depending on the results, the users can choose to repeat the training steps and the training data can be augmented with new labeled images to for example adjust the classifier to a new field situation. Additionally, a classifier can be trained only on a subset of the ground truth labeled images to support for example tuning of parameters with cross-validation.

4.4 Evaluation Criteria

In order to evaluate the plant classification pipeline, suitable evaluation criteria are required. Both a qualitative and a quantitative approach using performance metrics is defined.

The plant classification pipeline performs a classification task: The algorithm classifies image patches which are extracted at keypoint locations in the input image. The classifier outputs a classification score $\boldsymbol{s}_i$ for each keypoint k_i. Using the score vectors a receiver operator characteristic curve is plotted as defined in Section 2.2.4.

An evaluation on keypoint level, as done for the ROC curve, depends on the location and count of the keypoints. To be able to compare results of experiments with different numbers of keypoints (this happens for example when the patch size is changed), classification metrics are calculated on the interpolated plant classification image. These metrics then do not depend on the number and location of keypoints. They are implemented to compare each estimated vegetation pixel in the classified image to the ground truth image. Background non biomass pixel which are masked by the vegetation segmentation method in Section 3.2 are ignored. This avoids a bias for images with a lot of background pixels.

First, classification metrics are calculated before the smoothing process. The unsmoothed estimated labels l at each keypoint are interpolated to a full image of interpolated labels l_{interp} (using the presented interpolation processes while skipping smoothing). Using the classification metrics introduced in Section 2.2.4 all ground truth labels g are now compared to the corresponding l_{interp} for all vegetation pixels. All metrics derived from comparing the unsmoothed data to ground truth are annotated with the keywords *before smoothing*.

Second, a final analysis is possible when the full smoothed and interpolated plant classification image is considered: The ground truth labels g are compared to all corresponding smoothed and interpolated plant classification labels $\hat{l}_{\text{interp}}$ for all vegetation pixels in the image. These metrics are the main output of the pipeline and are annotated with the keywords *after smoothing*.

In addition to these quantitative evaluation metrics, also qualitative evaluation using visual inspection is performed. The evaluation and discussion of the plant classification pipeline is performed in Chapter 6.

4.5 Parameter Selection

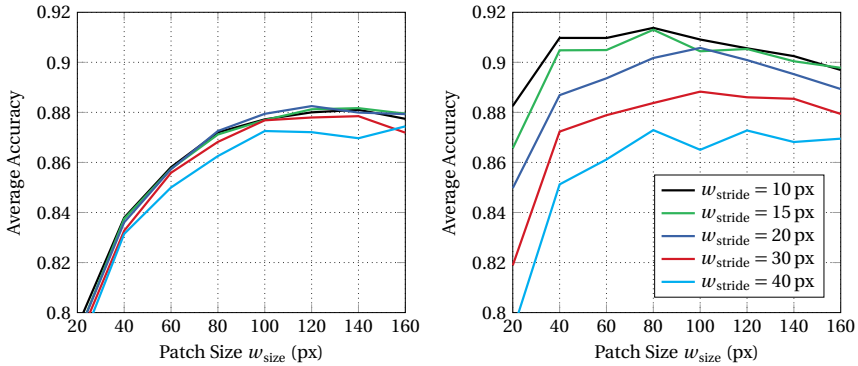
Now the complete plant classification pipeline has been presented. The goal of this section is to analyze which parametrization gives best results. The analysis of the pipeline with regard to the overall classification performance on different datasets is performed in Chapter 6 in detail.

Each pipeline run is performed using 5-fold cross-validation. To evaluate the performance of the pipeline the average accuracy measurement is chosen and evaluated on dataset A. The full plant classification image is used to calculate the metrics by comparing each ground truth labeled pixel to the estimated smoothed and interpolated plant class. The metric average accuracy and the dataset A are introduced and discussed in detail in Chapter 6. The focus of this section is to determine how the best parameterization can be determined and to understand the influence of each parameter.

4.5.1 Patch Size and Patch Stride

The main parameters of the patch extraction step are the patch size w_{size} and the patch stride w_{stride} between neighboring patches. In the following the impact of the patch size and patch stride on the classification performance is analyzed.

Figure 4.16 displays the average classification accuracy for different patch sizes w_{size} (20 px to 160 px) and strides w_{stride} (10 px to 40 px) before and after the smoothing step (see Section 4.2.4).



(a) Before smoothing **(b)** After smoothing

Figure 4.16: Average classification accuracy before (a) and after (b) smoothing for varying patch sizes w_{size} and patch strides w_{stride}.

The following trends can be observed:

1. With growing patch size the performance of the plant classification system before smoothing increases. Growing patch sizes result in more context being available in each patch, thus the classifier has more information to distinguish the different patches.

2. When the stride is increased, the performance generally drops. This is clearly the case for stride sizes larger than 20 px, for the stride sizes 10 px and 15 px there is no trend visible and the performance is roughly equal with the stride size 10 px.

3. The influence of stride size is largely independent from the choice of patch size. Both parameters can be individually chosen.

A comparison of the results before smoothing (Figure 4.16a) with the results after smoothing (Figure 4.16b) indicates that in general smoothing improves the classification accuracy. Very large or small patch strides do not benefit from smoothing as much as patch strides of 10 px or 15 px. Additionally, a performance maximum at patch sizes of around 80 px is visible, after that size the performance starts to drop slightly (but it is still significantly better than without smoothing).

Figure 4.17 displays the patch count for all of the patch stride and size combinations from Figure 4.16.

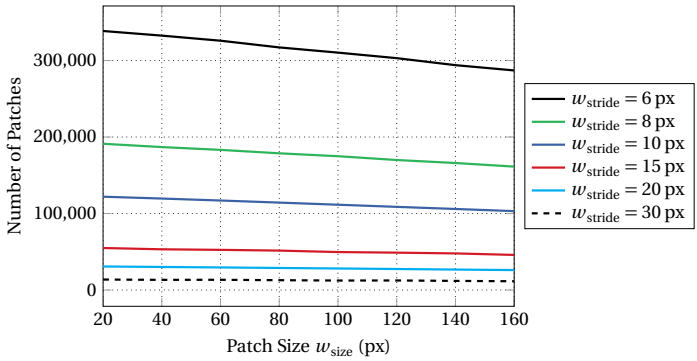


Figure 4.17: Number of patches depending on patch size and patch stride.

The declining number of patches for growing patch sizes and strides has two impacts:

1. The higher the number of patches, the higher the computational load.

2. The smaller the number of patches the less data a classifier has to learn from, especially when cross-validation and bagging (as in a Random Forest) happens.

Therefore, the patch size and stride should be chosen to achieve best performance while keeping the number of patches at a reasonable level.

Finally the following parameterization is chosen which balances the two requirements of enough training data and best performance:

- The default patch size w_{size} is set to 80 px.
- The default patch stride w_{stride} is set to 10 px.

This achieves best performance while retaining more than 110 000 patches for the 150 images, which is enough data for most classification algorithms.

4.5.2 Smoothing Parameters

The impact of choosing the smoothing parameter λ is displayed in Figure 4.18. It can be seen that when setting the parameter $\lambda = 0$ the same performance as without smoothing is achieved. This indicates that although inference using the CRF is performed, the performance — as expected — neither increases nor decreases.

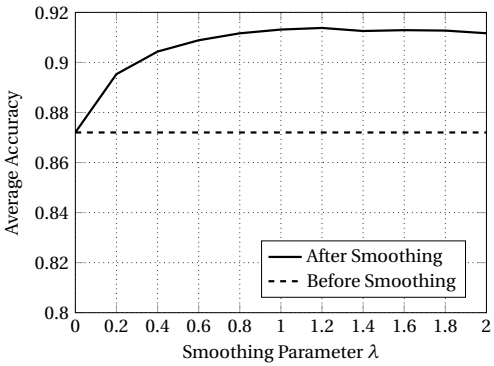


Figure 4.18: Average accuracy of the plant classification result depending on the choice of smoothing parameter λ.

The optimal parameter for λ is 1.2 while it can be noted that the plateau around the optimum is flat and not a sharp peak. This indicates that the smoothing process is robust regarding the concrete choice of λ.

4.5.3 Classifier Parameters

The classifier is one major component of the system. Machine learning algorithms also have parameters which need to be set depending on the data and application. Therefore, in the following the Random Forest classifier is analyzed in detail for the plant classification task.

A Random Forest classifier provides the internal out-of-bag error according to Section 2.2.5 as measure of classification performance. To determine the best parameterization for the Random Forest for the plant classification pipeline, a classifier is trained on the full dataset and its out-of-bag error is used to judge the performance of a specific parameterization.

The lower the out-of-bag error, the better the classifier. All experiments are repeated 3 times and the average out-of-bag and average training time are reported.

Number of trees The main parameter of a Random Forest is the number of trees which are grown. Figure 4.19 shows that the out-of-bag error of the classifier decreases with an increasing number of trees which are trained.

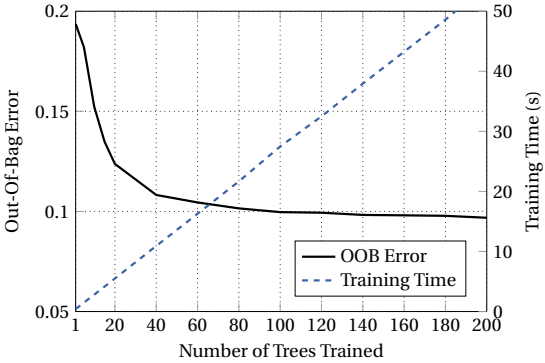


Figure 4.19: Out-of-bag error of the Random Forest classifier (left axis) depending on the number of trees it was used to train. Additionally, the right axis displays the time taken to train the classifier.

Additionally, the training time when using a single CPU core is plotted (right axis in Figure 4.19). It can be observed that the training time increases linearly with the number of grown trees. The out-of-bag error decreases sharply between 1 and 20 trained trees and then the error levels off and only decreases marginally. Training more than 100 trees results in only a minor improvement of performance while encountering the linear increase in training time.

Therefore, in the following the default number of trees is set to 100 to achieve a good out-of-bag error while not spending too much time on training.

Size of Leaf Nodes Figure 4.20 displays the out-of-bag error when varying the minimal size of a leaf node in the tree. It can be observed that with higher minimal leaf node sizes the error increases and the training time only decreases slightly.

The increase in out-of-bag error does not justify small savings in training time, thus the minimal leaf node size is set to 1. This results in fully trained trees with just a single pure label being stored in each leaf node.

Number of Features considered per Split Another important parameter is the number of features which are considered during each split when the classification tree is trained. Figure 4.21 displays the resulting out-of-bag error and training time when the parameter

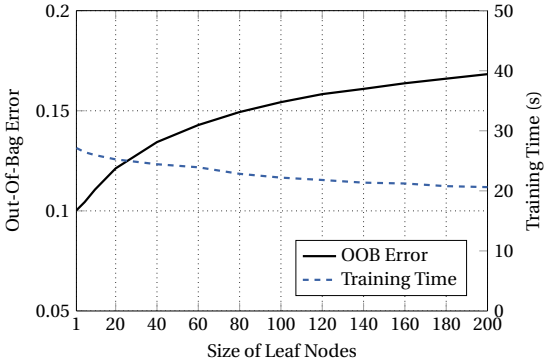


Figure 4.20: Out-of-bag error of the Random Forest classifier (left axis) depending on the leaf node size (minimum number of samples in each leaf). Additionally, the right axis displays the time taken to train the classifier. Note: The scales of the axes are kept in sync with Figure 4.19 to allow easy comparison.

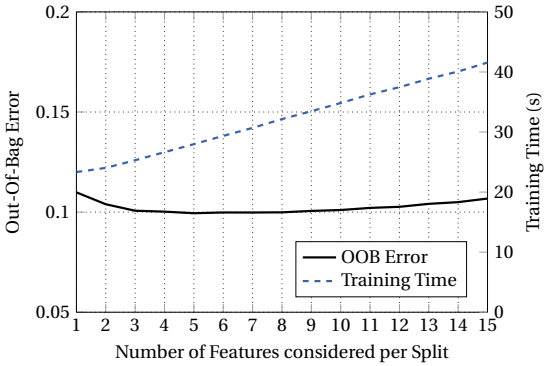


Figure 4.21: Out-of-bag error of the Random Forest classifier (left axis) depending on the number of features considered per split. Note: The scales of the axes are kept in sync with Figure 4.19 to allow easy comparison.

is varied. Here values of 1 to 15 are possible since the plant classification system extracts 15 features.

It can be concluded that the error is nearly flat for 4 to 9 features considered at each split while the training time increases in a monotone fashion. The recommended default value for a Random Forest is defined as the square root of number of features in the literature, here this would equal $\sqrt{15} = 3.87$. Finally, the parameter number of features considered

per split is set to 4. This is right in the area where the out-of-bag error is minimal, close to the default value and the training time is also kept as low as possible.

4.6 Summary

This chapter introduces a novel plant classification method for field images:

- The plant classification system is able to classify two or more plant classes in field images without the need for a plant or leaf segmentation.
- Feature extraction and classification are performed on overlapping image patches. This approach enables the pipeline to handle overlap of plants and irregular shaped leaves.
- The patch extraction at a sparse grid improves the runtime performance compared to full classification of all plant pixels. The newly introduced smoothing step compensates the loss of spatial output precision of previously known cell-based methods. The presented pipeline outputs a smoothed full plant classification image.
- Training of the system is done offline by an expert user. The classifier is then applied in the online phase.

Quantitative results and an evaluation of applying the novel plant classification system on two challenging field datasets is presented and discussed in Chapter 6.

5 Plant Position Estimation

An important property of a plant is its location in the field which can be described by the plant's stem position. Accurate detection of the position of a plant (Figure 5.1) enables precision plant treatment tasks including single plant weeding, grasping or fertilization.

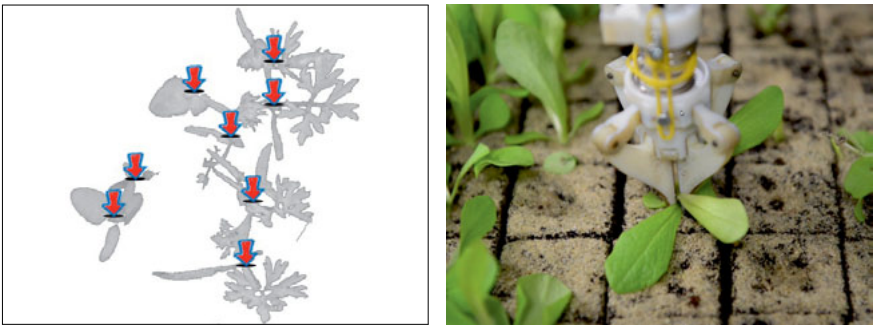


Figure 5.1: Field image with marked plant stem positions (left) and precision grasping of single plants based on stem position information (right).

The precision agriculture trend is continuously evolving due to ecological and economical reasons. This trend has the effect that farming tasks are developed where individual plants are treated compared to the previously applied methodology where whole fields or complete rows of plants are treated homogeneously.

Once agriculture tasks focus on single plants, the plant stem position is one of the most important features of a plant. It describes the center of a plant's activity and is the connection point of a plant's leaves and the root system. Therefore, the plant stem is the main attack point for single plant weed control. Treatment of plants at their stem is the most effective method for mechanical weed control [73] and other precision agriculture tasks.

This chapter presents a novel method for plant stem position estimation: The goal is a solution to the general plant position estimation task, where the stem positions of all plants (i.e. both crop and weed) in the image must be determined.

In the following the plant stem detection problem is analyzed in a very broad scenario: The required input data only consists of multispectral images from a downward looking camera. Prior segmentation into individual plants or leaves is not reliably feasible in

outdoor field environments (see Section 3.2). Therefore, plant or leaf segmentation is not a precondition for the new detection pipeline. This will allow the application of this system to real word field situations with many plant species and overlap between plants. Compared to constrained scenarios like greenhouses, in the field the plant position cannot be determined based on scene information like fixed field structures or detection of the pot position.

The presented objective of using only limited input data and not imposing additional constrains on the field scene is challenging. However, this ensures that the developed system is a generic solution that can be applied to a wide range of real world challenges in precision agriculture. This distinguishes the system from other approaches presented so far that heavily depend on special scenes or use additional information like GPS [46] or row context [54].

Throughout the chapter the plant position is defined to be the location where the plant stem emerges from the soil. Given the mounting position of the camera within the vehicle (extrinsic calibration) the plant position can be calculated in the robot frame (main coordinate frame of the robot). Using this position information, a treatment tool like the custom built gripper shown in the right picture in Figure 5.1 can be controlled to reach the plant position. Additionally, when the robots position is tracked over time or when the robots position is determined with external sensing like GPS or a localization system (SLAM) the plant position can be expressed in a fixed coordinate frame (for example a field coordinate frame or GPS coordinates.)

For the discussion in this chapter the term plant position is defined to specify the image coordinates u, v of the plant stem in the current 2D image. All further positions (derived from various transformations to different coordinate frames) is application specific and not in scope of this chapter.

In the following sections first related work is discussed, then the individual processing steps of the newly developed method are introduced in detail.

5.1 Related Work

The detection of plants and the estimation of their precise location has been studied from different perspectives: Some projects focus on the detection of plants or leaves in situation like homes or experimental setups; however these situations are substantially different from the agricultural use case of natural outdoor field environments.

In the following, the focus lies on work done in the agricultural domain with many plants growing in outdoor field or greenhouse farming applications. Depending on the sensing mode (2D image, 3D data, external sensing with for example GPS) and the information processing mode several approaches have been developed.

Row Detection Based Methods Row detection methods have been applied to detect the positions of row crops in fields [29, 53, 54]. The applied methods are mostly binarization

of the image and then either Hough transform related methods or filtering and fitting of row lines or polynomials [55] to previously extracted row center candidate points.

Some work has focused on even more constrained scenarios where the underlying spatial arrangement of the plants is to a large extent known in advance (for example when crops grow in a regular grid).

Søgaard and Olsen developed a computer vision system to detect crop rows in perspective field images [53]. The position of the crop row is calculated without segmentation, rather they use weighted linear regression on center of gravity calculation of image slices. This method only returns row position but not positions of individual plants.

The robotic system developed by Åstrand and Baerveldt applies a camera to guide a robot along a row of plants using Hough transform with 2 cm accuracy [29]. Based on the row they determine individual plants using a classification system based on color and shape features (crop with diameter of approximately 5 cm) but do not report plant detection results.

The automatic row detection approach from Jiang et al. proposes a computer vision pipeline which applies a multi region of interest approach to find crop rows based on candidate center points [54]. They claim that this approach is superior to Hough based methods and report a row detection rate of 93 % for images with 640 × 480 px.

These methods are not applicable in our use case: In the field the spatial distribution of all plants is not known, especially also the position of the irregularly appearing weed plants has to be estimated.

Plant Segmentation Based Methods To achieve a more precise plant position estimate, plant segmentation based methods were developed. The main idea is to segment the image into plants or leaves and then derive a plant position estimate from the segmentation [31, 56, 57, 58, 59]. The challenges for these approaches are the segmentation accuracy with high weed infestation or overlap between plants.

Kiani & Jafari segment field images into individual plants and derive the centroid position of each plant [58]. Their experiments with corn show that the centroid is a good estimate of the crop stem location. They attribute their good performance to the very different appearance of crop and weed when they performed their analysis. An evaluation of the stem detection accuracy for weed plants is missing in the paper.

Midtiby et al. perform leaf segmentation and then determine plant stem candidates by searching from the leaf tip [57]. The information of multiple leaves is fused to determine an estimate for the plant stem emerging point. Their system correctly identifies 90 % of plant stem emerging points correctly with a detection threshold of 2.0 cm.

Onyango & Marchant work with images of plants that are growing in a regular grid [31]. Their work focuses on crop/weed segmentation. However they estimate the plant position by calculating the centroid of the segmented plants or based on the soil pixels between plants and knowledge of the plant spacing. The studied crop is cauliflower with little overlap between adjacent plants (different growth stages are studied). They

do not report plant position performance scores and only do visual analysis of output images.

Huang & Lee present a vision-guided grasping system for phalaenopsis plantlets. Their experiments are however done in an artificial lab environment where plants are taken out of the soil and placed on a black background [56]. After segmentation of single plantlets through thresholding, the plantlet skeleton is calculated by thinning the binary segmentation mask. Then junction points of the skeleton are used to separate the plantlet into leaves and root. Finally, the grasping point is selected in the middle of the root and its 3D position is derived using stereo vision. A successful pickup rate of 78.2 % is achieved in an experiment with 348 plantlets. However, in field experiments this approach cannot be applied because the root is not visible when the plant is still growing in the soil.

Hunt et al. strive to estimate the position and area of ryegrass plants artificially planted in PVC rings arranged in a grid [59]. In the field of view of the camera an additional color calibration pattern is located. To determine the plant position, first vegetation pixels are selected through color calibration and thresholding. Second, edge detection is applied to the vegetation pixels and based on the detected edge lines plant centers are estimated (ryegrass has long leaves growing outwards). If this method does not yield any result, the PVC ring is detected in the image. This method is not applicable to naturally growing plants and does not handle large overlap between plants.

Furthermore, approaches which process side view image were proposed to find for example fruit or vegetables growing on large plants. Yamamoto et al. process side view images of plants [164]. They apply classification trees on 2D RGB color images to detect tomato fruits as blobs in the image. These methods are however not applicable to the desired use case of downward looking images.

This line of work relies on a prior segmentation of the image into plants or leaves. For in field situations this can be very challenging and errors from the segmentation process have a large impact on position estimation performance. Additionally, all papers have worked with large plants and scenarios with few weed plants present. In our challenging scenario with small crop plants and high weed infestation crop/weed segmentation does not work well (see Section 4.1) and a different approach is required.

Georeferencing Based Methods In addition to the image-based methods for plant position estimation GPS-based methods have been developed: High precision RTK GPS has been a driver in automation and field management in agriculture. With RTK GPS a positioning precision in the centimeter accuracy range can be achieved. This is used for automated vehicle guidance, yield mapping, etc. and can also be applied to plant position estimation [44, 45, 60]. The basic idea is to map the location of seeds during sowing and to use this global position information to find the plants in the field during later processing steps [61, 46]

Ehsani et al. equipped a four row planter with an RTK GPS unit and generated seed maps during and report deviation between the seed map and seed positions in the field

of 3.0 cm to 3.8 cm [44].

Griepentrog et al. attached an RTK GPS receiver onto a precision seeder [45]. In field experiments they were able to generate seed maps with 1.6 cm to 4.3 cm accuracy (depending on vehicle speed and seed spacing) and conclude that this is precise enough for vehicle guidance and potentially single plant precision agriculture.

Nørremark et al. use a similar approach to map seeds with RTK GPS when sowing sugar beet [60]. In addition to GPS data they used a tilt sensor and filtering to generate the field map offline. They show that 95 % of the seeds emerged up to 3.73 cm from the seed map when sowing with vehicle speed of 5.3 km/h. The follow-up work by Nørremark et al. presents a full GPS based plant mapping and detection system combined with a cycloid hoe which performs weeding [61]. The hoe has multiple fins and when the plant map indicates an upcoming crop area, the hoe is configured to avoid this space.

Sun et al. retrofitted a vegetable crop transplanter with RTK GPS and additional sensors to map transplanting crops to a field [46]. Then the position error of measured and real transplanted plants was analyzed. They report a mean plant position error of 2 cm with a 95 % error of 5.1 cm.

These georeferencing methods require expensive RTK GPS equipment and are limited by the positioning and sowing precision. They do not account for seed movement during sowing and can not cope with seeds that do not germinate. The largest restriction of these methods is however, that they cannot be used to detect the position plants that were not sown and mapped, which includes all weed plants! Additionally, the typically reported errors for crop plants are 2 cm to 5 cm and thus too high for the targeted precision agriculture tasks for high value crops.

A special approach is presented by Raja et al. and follows the same idea to re-detect the crop position in the field [165]. However, they do not apply seed mapping, they mark the crop plant while planting. They propose different methods and for example use paint (transplanted crop such as salad), plastic markers or genetically modify the seed such that the crop has fluorescent properties and can be re-detected with special camera equipment. This method has similar drawbacks as the georeferencing approaches and is not applicable to desired use case.

3D Sensing Based Methods Additionally, 3D sensing and processing has been applied to the problem [62, 63, 64, 65, 66, 67]. Some approaches process side view depth or stereo images of plants to detect stem and leaves. Others apply a lidar scanner instead of a camera to record a 3D pointcloud and then perform plant detection.

Nakarmi & Tang use side-view depth images to measure inter-plant spacing (stem to stem) in corn fields [63]. After preprocessing the side view images using filters and morphological operations, the side view stem skeleton is used to determine the plant position. They achieve 1.7 cm root mean squared error. The plants are very large and the stem fills almost the complete side view image, therefore the method is not applicable to downward looking images or small plants.

Weiss & Biber present a plant detection and mapping system using the FX6 3D LIDAR sensor [62]. Plants are identified in the 3D point cloud by removing ground pixels and clustering points according to a crop row model. The system does improve the plant position estimation by tracking the plant 3D points from scan to scan until they disappear from the field of view. They report position accuracies for plant detection of 3 cm in laboratory and 3.5 cm in field experiments with paper corn plant replicas.

Dey et al. developed a method to classify side-view images in vineyards into grapes, leaves and stems [64]. First, they fuse multiple images into a 3D view, then they apply saliency feature extraction, classification and smoothing of the reconstructed 3D points to extract grape wines, branches and leaves.

Bac et al. detect the stem of sweet-pepper plants using the support wire for the plants as visual cue [66]. However, their use case is different because a side view is used and the stem is located for the purpose of harvesting and not for locating point where the plant is growing out of the soil.

Alenyà et al. added a time-of-flight 3D camera to a robotic arm with a plant prober [65]. Then they used the 3D image data to extract suitable probing points on leaves of potted plants using surface modeling and graph based segmentation. Their approach however requires the robot to first move to a far away position to acquire and image, then a close up position to refine the segmentation. Additionally, the approach was shown for potted plants with large broad leaves, required good leaf contours which must be visible in the 3D image and the leaves must not move during image acquisition and probing.

Kusumam et al. present a 3D image processing pipeline to segment broccoli heads from two down looking pointclouds [67]. Using 3D features, KNN and SVM classifiers are applied with temporal smoothing to segment the 3D points belonging to the broccoli plant. Since the stem is not visible, this approach does not deliver a precise position and it is unclear if it is applicable to small plants with pinnate leaves where the applied 3D camera struggles. Furthermore, processing time is 5-6 s per image.

These methods rely on 3D sensing technologies and are are not applicable to 2D single-view camera images. The already presented methods mostly utilize side views of large plants and have limited precision. Furthermore, the 3D acquisition technologies in the discussed studies have low spatial resolution. This renders these methods not suitable for small crop plants and single plant weed control.

5.1.1 Summary of Related Work

The review of related work shows that plant position estimation research can be divided into four major approaches. First, row-based methods detect the crop row in the field using different computer vision techniques and then try to locate crops in the row. This is however not applicable if the spatial arrangement of crops is not known and cannot be used to detect positions of weed plants reliably. Second, plant segmentation can be applied

to obtain an estimate of a plant's position in the field. This only works for large plants without overlap and requires plant segmentation approaches, which is error-prone in field situations. Third, georeferencing methods were presented for plant position estimation by recording a plant's position while sowing or transplanting. These methods do not work for weed plants at all since they are not sown neither planted. Additionally, the precision of the georeferencing plant position estimation methods produces errors in centimeter range. Finally, newer approaches leverage 3D sensing of side views of plants to estimate their position. Such approaches require special sensors and still have limited precision.

The newly developed plant detection and position estimation pipeline of this thesis overcomes these limitations and is able to process single view 2D images of small overlapping plants (both crop and weed) in outdoor fields. The approach is able to deliver plant position detection with the desired precision of less than 0.75 cm deviation from ground truth (see Section 6.3). The proposed method neither requires row information nor a plant segmentation. Additionally, the position of all plants is estimated, the restriction to only produce position estimates for crops of for example the georeferencing methods does not apply.

5.2 Novel Plant Position Estimation Pipeline

In the following a novel plant position estimation pipeline is developed and evaluated using real world datasets. The plant position estimation system comprises an online and offline process which both apply different computer vision and machine learning techniques. Figure 5.2 gives a detailed overview of the different processing steps from the input data to the plant stem detection result.

The online stage processes unseen images with the goal of detecting plant stems: A sliding window is applied and each image patch is classified whether it displays a stem or non-stem region. From these classification scores, postprocessing steps estimate the plant stem positions.

The offline training process with human interaction creates a ground truth database, which is used to train the machine learning part of the system. Some processing steps from the online part are reused, however for example the patch extraction step is conducted in a slightly modified manner.

In the following the pipeline is discussed in detail. First, the online steps, then the offline steps are described in detail.

5.2.1 Patch Extraction during Online Phase

Stem detection and position estimation calculations are performed on small image patches, which are extracted from the field image where the soil background is masked (see Sec-

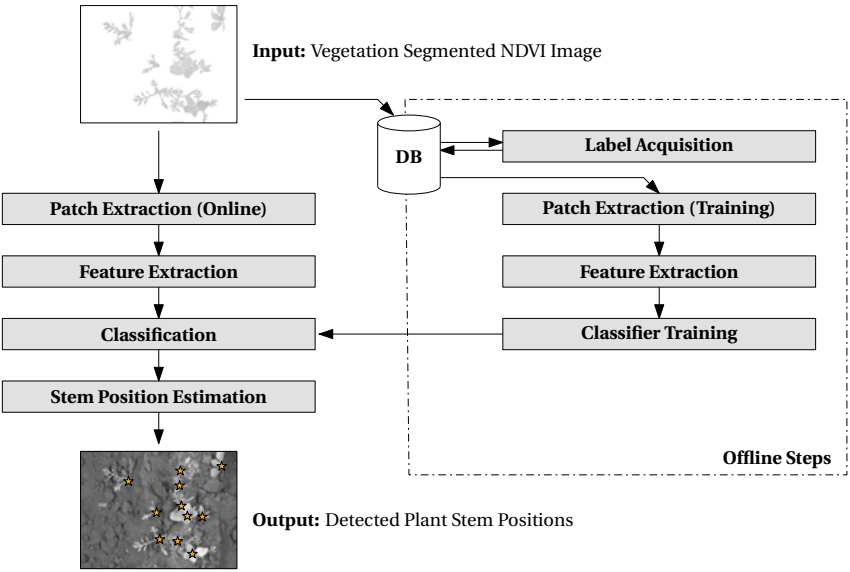


Figure 5.2: Processing steps of the plant position estimation approach. The pipeline processes the segmented NDVI images into plant stem positions. The processing steps comprise an online (left) and offline (right) stage. All processing steps are introduced in detail in the remainder of the chapter.

tion 3.2). The image patches are generated by applying a sparse sliding window to the image.

The exact methodology of patch extraction differs between the *online application* phase and the *offline training* phase. In the following, the patch extraction step is explained in detail for the online phase when new unseen images are processed. Figure 5.3 displays this graphically. The specific adjustments for the offline training phase are described below in Section 5.3.2.

When a new image is run through the trained pipeline, a sparse sliding window approach is applied: At every window position where biomass is located at the window center, an image patch is extracted. Once the patches are extracted, all patches and their patch position are forwarded to the next step in the processing pipeline. The center of a window position where a patch is extracted is called keypoint.

Figure 5.4 displays several exemplary patches extracted from the real world dataset where the ground truth label is represented by a colored border (green denotes patches from a stem region, red denotes patches from non-stem regions).

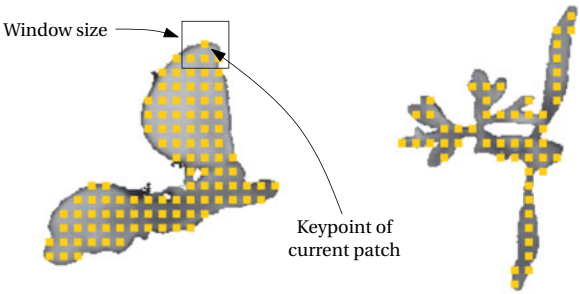


Figure 5.3: Patch extraction process during *online* phase: Patches are extracted using a sliding window approach. In the image all keypoint positions where a patch is extracted are visualized with a yellow dot.

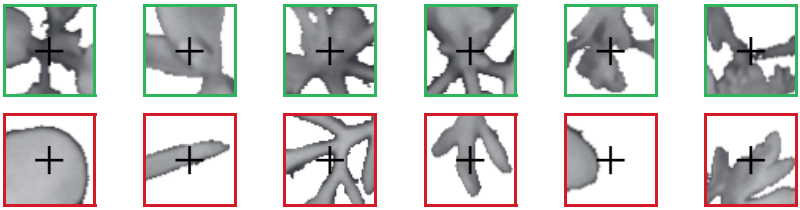


Figure 5.4: Example patches extracted for plant stem position estimation: The examples in the top row are positive stem patches whereas patches in the bottom row display non-stem regions. The black + indicates the keypoint of the patch.

5.2.2 Feature Extraction

The feature extraction step has the goal to define a suitable numerical representation which is useful to discriminate patches displaying the stem region from patches displaying other parts of plants. For plant position estimation 12 features are developed and defined in Table 5.1.

The feature vector $\boldsymbol{f}$ comprises 8 statistical and 4 geometrical features. The statistical features describe the appearance of the stem patch: The minimum, maximum, range, median, mean, standard deviation, skewness and kurtosis of biomass pixels in the NDVI image patch is determined. These features are similar to the plant classification features and make use of the different appearance of the NDVI patches if they display the stem region or not (as shown in Figure 5.4).

The geometrical features comprise the distance of the center of gravity (of biomass pixels in the patch) from the patch center, the mean and standard deviation of the distance of every biomass pixel from the center of gravity and area of biomass in patch. The first three

Table 5.1: List of features for plant stem position estimation.

f_i	**Description**
f_1	Minimum of Biomass Pixel Intensities
f_2	Maximum of Biomass Pixel Intensities
f_3	Range of Biomass Pixel Intensities
f_4	Mean of Biomass Pixel Intensities
f_5	Median of Biomass Pixel Intensities
f_6	Standard Deviation of Biomass Pixel Intensities
f_7	Kurtosis of Biomass Pixel Intensities
f_8	Skewness of Biomass Pixel Intensities
f_9	Distance of Center of Gravity (COG) from Patch Center
f_{10}	Mean of Distances from COG to all Biomass Pixels in Patch
f_{11}	Standard Deviation of Distances from COG to all Biomass Pixels in Patch
f_{12}	Area of Biomass in Patch

geometrical features describe how the biomass pixels are distributed around the center of gravity and the area describes the amount of biomass located in a patch.

Figure 5.5 displays the feature importance for each feature. The mean decrease of average accuracy is selected as feature importance measure and calculated on a fully trained Random Forest [157]. For each feature, the feature importance value is calculated separately by evaluating the average accuracy drop on the out-of-bag data when the respective features values are randomly permuted. For this evaluation dataset B is used which will be introduced in Section 6.1. It can be concluded that the features f_7, f_8 and f_1 contribute the most while feature f_{11} is the least relevant.

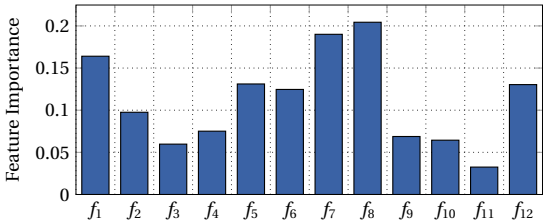


Figure 5.5: Feature importance of the plant position estimation features on dataset B. The feature importance (y-axis) is the mean decrease in accuracy when f_i is permuted during calculation of the out-of-bag error.

5.2.3 Classification

The classification step comprises the discrimination of patches displaying the stem region from patches displaying other parts of plants. During application of the pipeline an already trained Random Forest is used (see Section 5.3.3 regarding classifier training). All feature vectors (describing the patches to be classified) are fed into the Random Forest and the normalized score vectors $\boldsymbol{s} = (s_0, s_1)$ with $\sum s_i = 1$ are output. Then only the score value s_1 of the position class is taken and called stem score s.

The result of the classification step is a list of stem score values s (0 to 1) for all patches in the image. The scores can be plotted at the corresponding keypoints superimposed onto the image; see Figure 5.6 for such an exemplary stem certainty plot. These stem score values s in the stem map are used to derive the stem position in the next step.

Figure 5.6: Stem certainty map in color code from red (i.e. no stem, $s = 0$) to green (i.e. stem, $s = 1$). The scores are plotted at the corresponding keypoints.

5.2.4 Stem Position Estimation

The goal of the stem position estimation step is to postprocess the stem certainty map and to derive discrete stem positions. Key steps in this process are filtering of the stem certainty scores and non-maximum suppression to generate estimated plant stem positions.

For these filtering and non-maximum suppression steps, first a stem score matrix $\boldsymbol{S}$ is constructed from the list of stem score values s. The stem score matrix has as many

elements as there are window positions from which the patches were extracted. Using the sliding window arrangement it is possible to calculate the exact keypoint for each element of the matrix $\boldsymbol{S}$, which can be used for plotting or generation of the final stem detections.

Stem Certainty Filtering The stem certainty matrix $\boldsymbol{S}$ is filtered with a Gaussian kernel to smooth the individual stem certainty values. This step introduces the parameter k_{smooth} which is the radius of the smoothing kernel (described in Section 5.5.3). The result of smoothing the scores can be seen in Figure 5.7. The filtering step results in a more smooth classification which improves the effectiveness of the following non-maximum suppression step.

Figure 5.7: Filtered stem certainty map in color code from red (no stem) to green (stem). The image is the same as in Figure 5.6.

Non-Maximum Suppression From this smoothed stem certainty matrix $\hat{\boldsymbol{S}}$, discrete stem position are derived using non-maximum suppression. The goal is to not take each local maximum from the stem certainty matrix, but to suppress maxima which are close together. This follows the idea that in the real world plant stem centers are not infinitesimally small and/or very close together.

The result of non-maximum suppression can be seen in Figure 5.8 which displays the same image from Figure 5.7 where instead of plotting the certainty the estimated stem positions (u, v) are visualized.

The non-maximum suppression step applies a square suppression kernel of size $k_{\text{non\_max}}$. The kernel only responds with 1 if the center pixel of the windows corresponds to the maximum value of all values in the window. Otherwise 0 is returned. The kernel is convolved

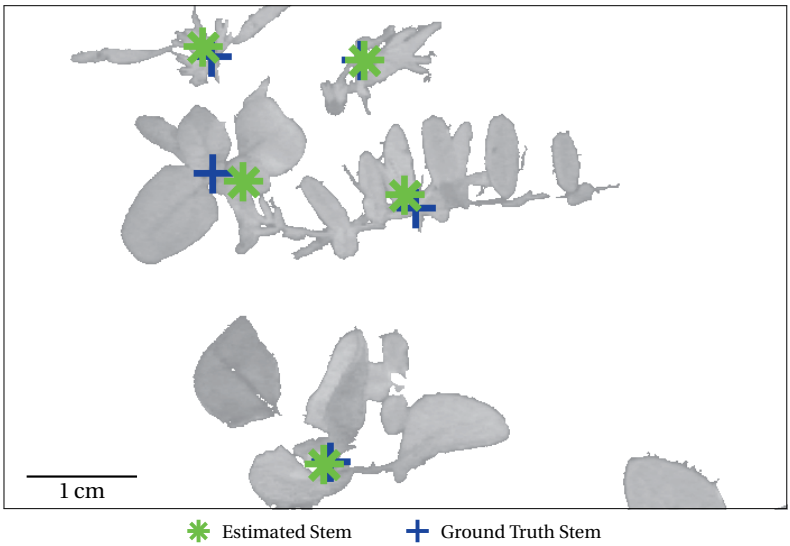


Figure 5.8: Visualization of the estimated stem positions (green stars) after non-maximum suppression. Ground truth stem positions are marked with blue pluses.

over the smoothed stem certainty map $\hat{\boldsymbol{S}}$. The resulting binary stem detection matrix is then converted into stem positions by transforming each entry in the matrix which corresponds to 1 into image coordinates u, v.

5.3 Training Phase

The training phase comprises all steps which are necessary to train the classifier. This comprises data acquisition and ground truth labeling, preprocessing of the labeled images into feature vectors and finally the training step of the classifier.

5.3.1 Ground Truth Data Acquisition

An important pre-requisite for supervised machine learning is the availability of ground truth labeled data. For the stem detection algorithm the ground truth stem positions are acquired manually by a human expert. Additionally, the class of the plant can be defined as meta data when a plant position is defined.

The data is collected by presenting the human a segmented image (soil pixels masked to white) in a web-based interface. The user then marks the stem center points graphically.

Figure 5.9 displays one screenshot of the labeling process. The web interface is based on the LabelMe tool from Russel et al. [163].

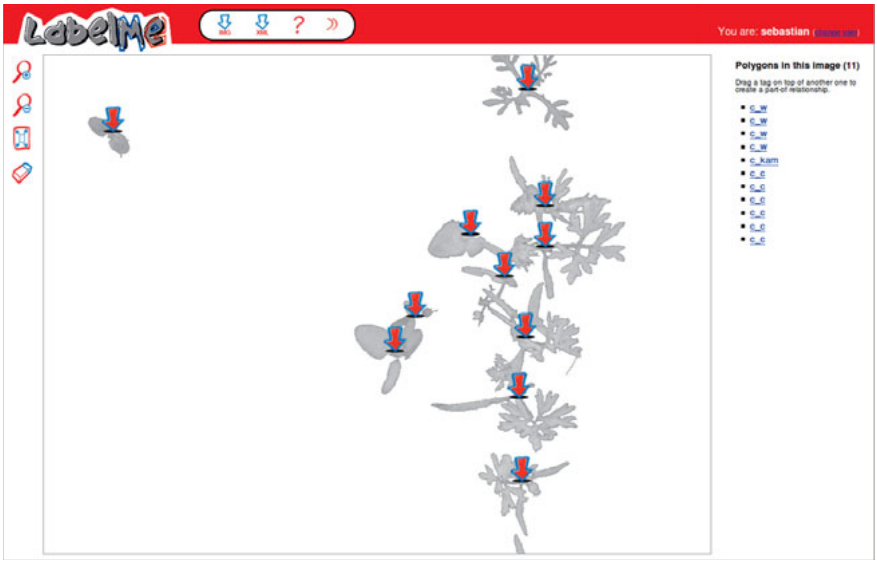


Figure 5.9: Labeling tool used to acquire ground truth data. A user marks the plant stem positions in the image with the mouse and places a marker (arrow). Additionally, a class can be associated to each marker. For broadleaved plants the stem is easier to detect than for carrot plants with pinnate leaves: There, the two elongated true leaves are a good hint to find the plant stem.

5.3.2 Patch Extraction during Training Phase

During training, patches are not just extracted using the sparse sliding window pattern with a fixed stride between adjacent patch positions. The labeled ground truth stem positions are taken into account to sample positive (stem region) and negative (non-stem region) patches. Figure 5.10 explains the patch extraction step for the training phase visually.

The plant position ground truth is available as discrete pixel position for each plant center and not as a full image or mask. Therefore, the sparse sliding window patch extraction process is suppressed in proximity of a ground truth stem position to form a border. A parameter d_{border} defines the square border region by using the Chebyshev distance metric. All patches extracted this way do not display plant centers and thus form the set of negative training examples (red boxes in Figure 5.10).

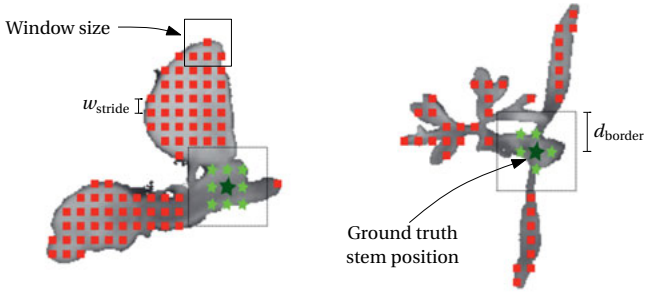


Figure 5.10: Patch extraction process during *training* phase: Negative (non-stem) patches are extracted at the grid locations marked by the red box. At and around the manually labeled ground truth stem position (dark green star) positive patches are sampled. These positions are marked by green stars.

Positive training examples are extracted around the plant stem positions. In a simple approach one could only sample at the ground truth stem position, however this leads to very few positive samples compared to many negative (non-stem) samples. Therefore and to also account for slightly imprecise ground truth stem positions additional positive patches are sampled in the proximity of the ground truth stem label.

The details on how the positive patches are sampled is visualized in Figure 5.11. First, the parameter d_{max} defines the maximum Chebyshev distance in pixels between an additional keypoint for s positive patch and the ground truth stem position. Second, the parameter d_{step} defines the step size in pixels in which patches are sampled within d_{max} both horizontally and vertically.

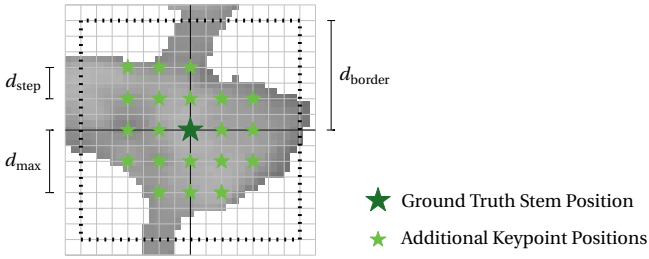


Figure 5.11: Details on how additional positive patches are sampled in the proximity of the ground truth stem using the parameters d_{max} and d_{step}. In the example $d_{\text{max}} = 4$ and $d_{\text{step}} = 2$ is selected. Note that d_{border} is normally selected much larger compared to d_{max} to ensure that positive and negative patches do not overlap.

The patch extraction step during the offline training phase introduces the following parameters:

1. The patch size w_{size} and stride w_{stride}. Without loss of generality, a square patch and an equal stride in horizontal and vertical direction are assumed. Additionally, the patch size and stride are kept at the same values during the online and offline phase.
2. The length d_{border} defines a square border (Chebyshev distance) around a ground truth stem position in which no negative patches are sampled.
3. The sampling pattern of additional positive patches which are sampled around each ground truth stem position are determined by the distances d_{max} and d_{step} as explained above.

5.3.3 Classifier Training

During classifier training, the feature extraction step is performed (as specified in the section above) on labeled images. Then all extracted feature vectors with ground truth labels (stem or non-stem region) are used to train a classifier. Here a Random Forest is trained, see Section 2.2.5 for details on the Random Forest algorithm. The classifier is then saved and can be used in the online application phase to classify new unseen patches into the desired stem or non-stem classes.

5.4 Evaluation Criteria

A key step of any machine learning system is a well defined evaluation procedure with suitable ground truth data and relevant metrics. The developed stem detection pipeline performs a detection task: The algorithm estimates stem points represented by image coordinates $\boldsymbol{p} = (u_{\text{p}}, v_{\text{p}})$. Ground truth data is available and encoded as ground truth stem point $\boldsymbol{g} = (u_{\text{g}}, v_{\text{g}})$.

Given that the data type (for both the detected and ground truth stem) is a point in the two dimensional space, an euclidean distance based error metric is chosen. The indices u and v determine the image coordinates for either the detected point p or the ground truth point g.

$$E(\boldsymbol{p}, \boldsymbol{g}) = \left|\boldsymbol{p} - \boldsymbol{g}\right| = \sqrt{\left(u_{\text{p}} - u_{\text{g}}\right)^2 + \left(v_{\text{p}} - v_{\text{g}}\right)^2} \tag{5.1}$$

The quality of a plant stem detection is then evaluated by setting a detection threshold γ, which defines the acceptable euclidean distance between the ground truth and detected stem location. If a point satisfies Equation (5.2) the detection is considered to be correct.

$$E(\boldsymbol{p}, \boldsymbol{g}) < \gamma \tag{5.2}$$

The next step is the assignment of the detected plant stem positions to ground truth positions. Each assignment shall fulfill Equation (5.2) and we strive to find the optimal assignment of detected and ground truth positions. This is known as optimal assignment problem and can be solved with the Hungarian Method [166]. All entries in the cost matrix for which Equation (5.2) is not satisfied are set to infinity to avoid an assignment in this case. The result of the assignment step is a list of assignment tuples $(\boldsymbol{p}, \boldsymbol{g})$ which fulfill the stem detection threshold γ.

After assignment of detections to ground truth positions, valid assignments are counted as hits. Additionally, ground truth positions without assigned detections are counted as misses and detections without assigned ground truth positions are counted as false alarms:

Hit Stem detection valid. Fulfills stem detection criterium in Equation (5.2).

Miss No detected stem position could be assigned to a ground truth stem position.

False Alarm No ground truth position could be assigned to this detected stem position.

Using the hits, misses and false alarms, a confusion matrix Figure 5.12 is created. The confusion matrix only has three entries which are filled with hits, misses and false alarms. The entry for true negatives is set to zero because it does not exist for the detection task.

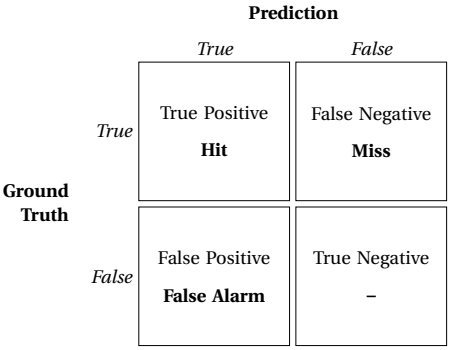


Figure 5.12: Confusion matrix for stem detection. Hit, miss and false alarm map to true positive, false negative and false positive respectively. True negatives do not exists due to the nature of the detection task.

From the confusion matrix the following detection metrics are derived: precision, recall and F1-score according to Equations (2.4) to (2.6).

In addition to quantitative evaluation visual inspection can be applied as a qualitative evaluation technique. For example the segmented field image is plotted in Figure 5.13 with detected and ground truth stems together with information which detections are hits, misses or false alarms. This visual representation is a good measure to evaluate the

performance of the system and to detect issues, when for example the stem of specific plants cannot be reliably detected.

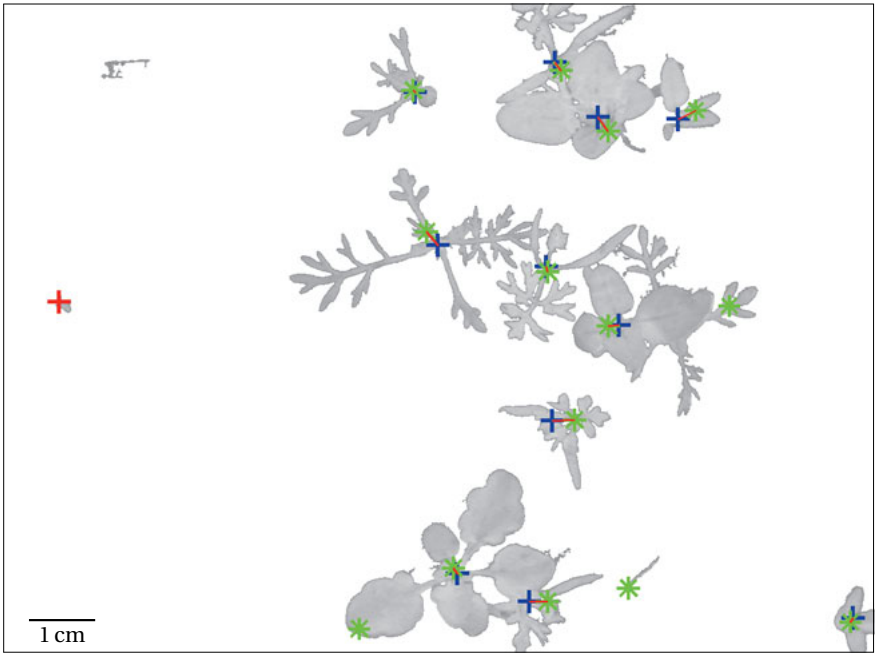


Figure 5.13: Visual inspection of the plant stem position estimation. Green stars mark detected stems, blue pluses ground truth locations for which a detection was found and red pluses make missed ground truth plant stems.

These three sets of evaluation metrics are used to evaluate the plant position estimation pipeline in Chapter 6.

5.5 Parameter Selection

The goal of this section is to study the influence of the different parameters and to determine the parametrization which gives best results. The analysis of the pipeline with regard to the overall stem detection performance on different datasets is performed in Chapter 6 in detail.

To evaluate and determine parameters with best performance the F1-score metric is chosen. It balances precision and recall and thus forces a parameterization which strives for a good detection rate (many and precise hits) while not allowing too many false alarms. If not specified otherwise the results are generated using 5-fold cross-validation.

Here dataset A is used and the stem detection threshold γ is set to 68 px which equals approximately 7.5 mm. Both the dataset and selection of γ are introduced in depth in Chapter 6. If not specified otherwise the default parameterization as given in the section above is used to produce the results. The focus of this section is to understand the influence of each parameter and how the best parameterization can be determined.

5.5.1 Patch Size and Patch Stride

The plant position estimation pipeline applies a sliding window to extract patches. The two main parameters are the patch size w_{size} and patch stride w_{stride} (the amount of pixels the window is moved between subsequent patches). Figures 5.14 and 5.15 show the impact of varying the patch size and patch stride.

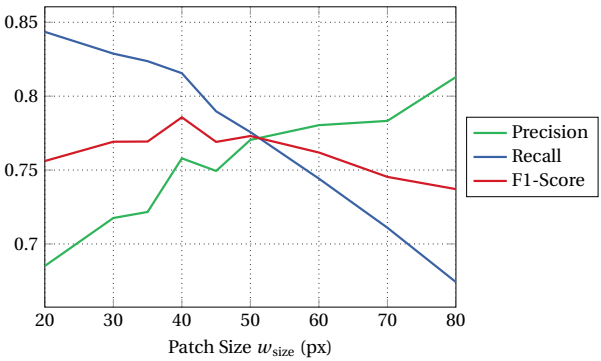


Figure 5.14: Variation of the patch size parameter w_{size}.

The plant position estimation performance F1-score metric exhibits a clear maximum at a patch size w_{size} of 40 px. When analyzing the patch stride parameter it can be observed that a patch stride w_{stride} of 10 px delivers the best F1-score.

The choice of patch size and patch stride has a great impact on all further processing steps as it influences the number of patches which are extracted. Therefore, the variation of patch size and stride are also analyzed again when both are varied together in Figure 5.16.

This plot supports that the choice of parameterization defined above yields best results. Large strides hurt the performance more than slightly larger or smaller patch sizes.

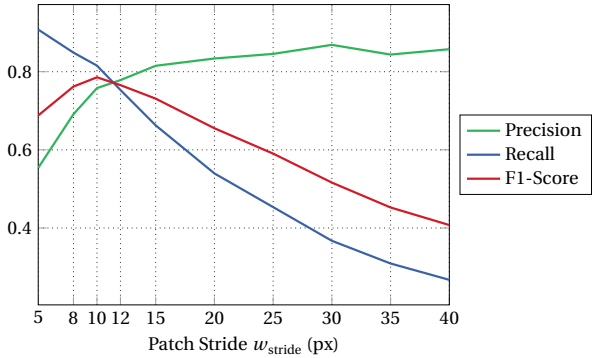


Figure 5.15: Variation of patch stride parameter w_{stride}.

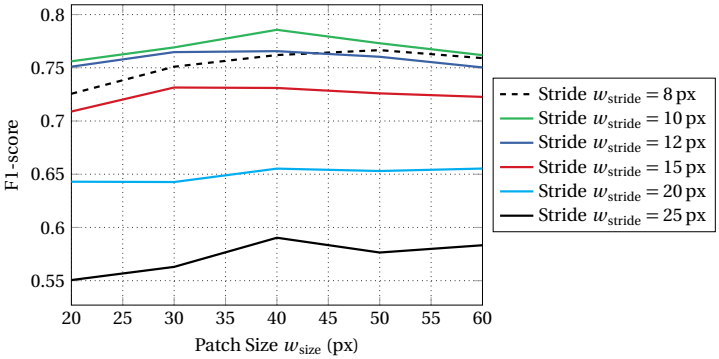


Figure 5.16: Joint variation of patch size w_{size} and patch stride w_{stride} parameters.

5.5.2 Training Patch Extraction Parameters

The patch extraction step during the training phase applied a modified sliding window approach, where not all positions are sampled equally. Depending on the labeled ground truth stem position, a border area is placed around each stem position where no negative samples are extracted using the sliding window. Then inside this region in the proximity of the ground truth stem positive training examples are sampled. See Figure 5.10 for an illustration of the scheme.

Border around Ground Truth Stem Positions This border around the ground truth stem positions introduces the single parameter d_{border} which specifies the distance from each ground truth stem where no negative patches are extracted. Here, the Chebyshev distance is utilized to achieve a square border region. Figure 5.17 displays the effect of

varying the parameter d_{border} on plant position estimation performance.

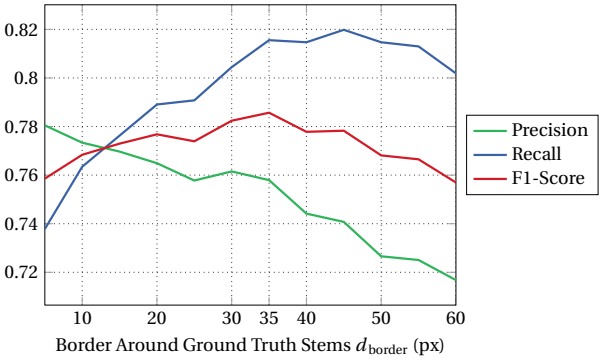


Figure 5.17: Variation of the parameter d_{border} specifying the border around each ground truth plant position where no negative patches are sampled.

The choice of this parameter exhibits a maximum F1-score at 35 px. Therefore, the parameter border around ground truth stem positions d_{border} is set to 35 px.

Training Patch Extraction Parameters The locations where positive patches are sampled in proximity of the ground truth stem position are defined by additional parameters. First, a patch is extracted right at the ground truth position. Second, additional patches are extracted in a grid in the proximity of the labeled ground truth stem position. All details on how and why patches are sampled this way are given above in Section 5.3.2.

The sampling step size d_{step} together with a maximum Chebyshev distance d_{max} (maximum distance between additional positive patch positions and the ground truth stem position) define the locations of additional extraction points as visualized above in Figure 5.11. These two parameters are jointly varied in Figure 5.18 while analyzing the resulting plant position estimation F1-score metric.

From the figure, it becomes clear that a larger maximum distance d_{max} is beneficial for maximizing the F1-score. The sampling step d_{step} influences how fast the maximum F1-score is achieved.

In order to not extract too many patches while achieving a high F1-score a sampling step size d_{step} of 2 px and a maximum Chebyshev distance to ground truth stems d_{max} of 12 px are chosen.

Lowering the step or increasing the sampling window creates much more patches and does not improve the score. This is explainable since sampling with a small step yields many patches which look similar. Increasing the sampling window will create positive patches which are extracted further away from the ground truth stem position which is not beneficial for training a classifier which is supposed to detect the stem region only.

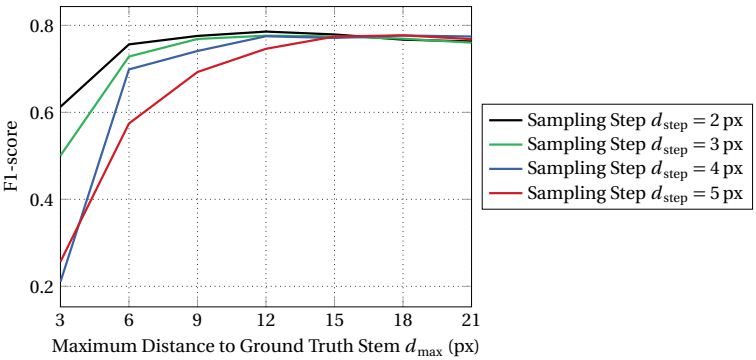


Figure 5.18: Joint variation of the sampling step d_{step} and the maximum Chebyshev distance $d_{\max}$ where positive patches are extracted in the proximity of ground truth stems.

5.5.3 Stem Position Estimation Parameters

The stem position estimation part of the pipeline comprises the two steps of smoothing the classifier output with a Gaussian kernel and subsequent non-maximum suppression.

Smoothing Parameter The smoothing parameter determines the size of the circular smoothing kernel k_{smooth} which is applied to the stem certainty matrix $\boldsymbol{S}$. Figure 5.19 displays the influence of this parameter on the plant stem position estimation F1-score. The analysis exhibits a clear maximum F1-score when setting the smoothing kernel size k_{smooth} to 3.

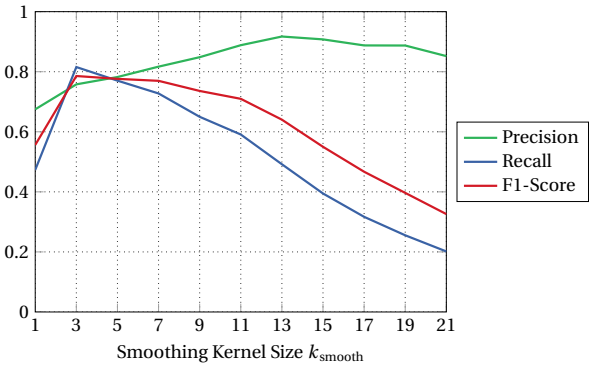


Figure 5.19: Influence on plant stem detection metrics for different sizes of the smoothing kernel k_{smooth}.

Non-maximum Suppression Parameter The second parameter which influences the stem detection is the square non-maximum suppression kernel size $k_{\text{non\_max}}$. Figure 5.20 displays the performance of the plant stem position estimation pipeline for different non-maximum suppression kernel sizes.

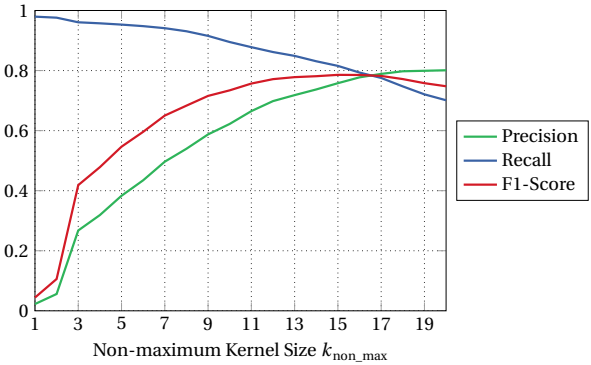


Figure 5.20: Variation of the non-maximum suppression kernel size $k_{\text{non\_max}}$.

The plot indicated that a clear maximum for the F1-score exists for a non-maximum suppression kernel size $k_{\text{non\_max}}$ of 15.

A broad plateau in F1-score is achieved for the chosen non-maximum suppression kernel size. When this plot is studied further it can be noted that $k_{\text{non\_max}}$ is well suited to balance between precision and recall if desired. The whole spectrum from a recall of approximately 1.0 to a precision of up to 0.8 can be reached by varying the kernel size.

5.5.4 Classifier Parameters

The plant stem detection system also applies a classification step with a supervised Random Forest algorithm. Therefore, similar to the plant classification pipeline the Random Forest's parameters are tuned.

However, in this case the final output of the plant detection pipeline is not suitable to tune these parameters. The classification is applied as internal step of the detection pipeline and the classifier's output is processed with non-maximum suppression. Therefore, in order to tune the classifier the out-of-bag error of the Random Forest is used instead of a plant position estimation metric.

All data in this section is generated by training a classifier on the full dataset while calculating the out-of-bag error according to Section 2.2.5.

Number of trees First, Figure 5.21 analyzes the out-of-bag error when the number of trees grown is increased. The training time on a single CPU core is plotted additionally as right axis.

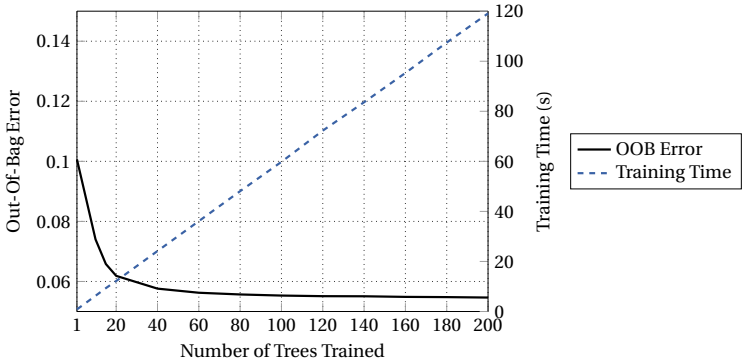


Figure 5.21: Out-of-bag error when the number of trees grown is increased (left axis). The right axis plots the training time spent to train the Random Forest classifier.

It can be observed that the out-of-bag error sharply decreases between 1 and 60 trained trees. Then the error continues to drop only slowly. The training time increases linearly in relation to the number of trees trained. The chosen number of trees of 120 is a balance between a good out-of-bag error and low training time. If lower training times are desired, the number of tree trained can be reduced at only marginal loss of performance.

Size of Leaf Nodes The minimal size of a leaf node in the tree is another parameter of the Random Forest which can be tuned. Figure 5.22 displays the out-of-bag error when this parameter is varied. Increasing the minimal leaf node size increases the out-of-bag error while the training time slightly decreases.

Therefore, the minimal leaf node size is set to the default value of 1 which implies that a single label is associated with each leaf node. A slight decrease in training time is not worth the reduction of performance.

Number of Features considered per Split Figure 5.23 displays the out-of-bag error and training time when the number of features which are considered during each split is varied. The range for this parameter spans 1 to 12 because the plant position estimation pipeline applies 12 features.

The out-of-bag error is minimal in the range of 3 to 5. Since the training time increases with higher numbers of features considered, lower numbers are preferred. Therefore the final Random Forest parameter number of features considered at each split is set to 3. This is close to the default parameter of $\sqrt{12} = 3.46$

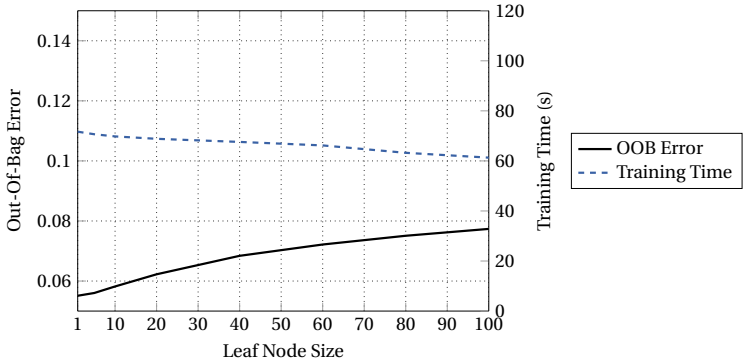


Figure 5.22: Out-of-bag error of the Random Forest classifier (left axis) depending on the final node size in each tree's leaf node. Additionally, the right axis displays the time taken to train the classifier. Note: The scales of the axes are kept in sync with Figure 5.21 to allow easy comparison.

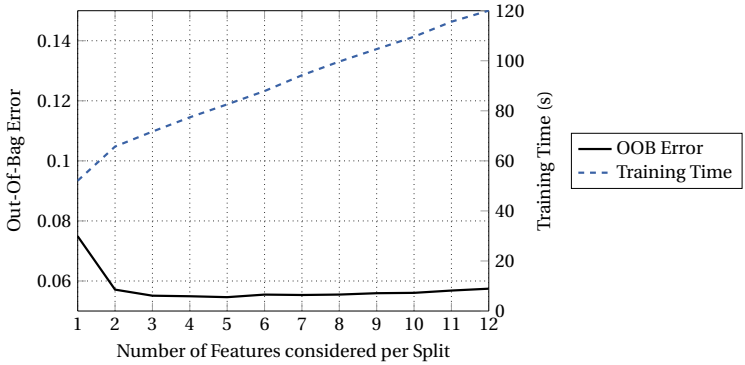


Figure 5.23: Out-of-bag error of the Random Forest classifier (left axis) depending on the number of features considered per split and training time of classifier (right axis). Note: The scales of the axes are kept in sync with Figure 5.21 to allow easy comparison.

5.6 Summary

The novel methods for plant position estimation developed in this chapter are:

- Plant position estimation is formulated as a detection problem. A sliding window-based classification approach is combined with non-maximum suppression to determine plant stem positions.
- Only downward looking images of plants are processed, additional information from for example GPS or row segmentation is not required. Still, the pipeline is able to estimate the position of both crop and weed plants.
- The stem detection method does not require error-prone segmentation of plants or plant structures like for example leaves or veins to derive the stem position. Instead, the local appearance of the plant stem region is used to train a classifier which is able to detect the stem in unseen test images.
- The proposed solution has the advantage that it is applicable to real world field images, where plants grow close together and overlap. The novel plant position estimation pipeline copes well with these field situations, works for all plant types and thus enables precision agriculture tasks like single plant weed control.

The next Chapter 6 consists of a thorough evaluation of the plant classification system (from Chapter 4) and the plant position estimation system presented in this Chapter.

6 Experimental Results and Discussion

This chapter provides the evaluation of the presented novel methods of Chapter 4 and Chapter 5 together with a discussion of the results.

First, the construction of a field robot and data recording of two new real world datasets are presented. Second, the plant and stem detection pipelines are evaluated on these datasets. This is followed by the discussion of the performance of the developed methods and the applicability of the plant classification and the stem detection pipeline to solve the precision agriculture tasks defined in the objectives of this thesis.

6.1 Data Acquisition Robot and Dataset Properties

In order to evaluate the new plant classification and position estimation methods appropriate data is required. Because of a lack of publicly available datasets two new reference datasets were recorded. In order to acquire data in a realistic and systematic manner, a field robot was built and equipped with a camera system for the recording of the datasets. Additionally, to foster research in this domain, one dataset was made publicly available in conjunction with a publication [70].

The construction of the field robot is described in the first part of this section, the recording and properties of the datasets are introduced in the second part of this section.

6.1.1 Field Robot for Data Acquisition

The so-called Bonirob is a multi-purpose field robot concept [167] which was developed for precision agriculture. The robot by itself does not implement a specific agricultural function, rather if features a utility bay where application modules (called apps) can be mounted.

During the research for this thesis a novel version of the Bonirob was developed with partners in the publicly funded project RemoteFarming.1 [72]. The Bonirob V2 field robot (Figure 6.1) is used throughout the experiments to acquire the necessary field image data. Additionally, the platform is used as demonstrator platform for the weed control process.

The Bonirob is a mobile robot with four wheels and omni-directional drive. All four wheels can be individually steered and are actuated (8 electrical drives). Additionally,

Figure 6.1: Two Bonirob V2 robots in a field: The robot has free space inside which can be used to mount an app module (right robot). The sensing and plant classification and position estimation system developed in this thesis is mounted in the left robot.

the wheels are mounted to legs (orange in Figure 6.1) which enable adjustment of the track width of each individual wheel. This allows the robot to navigate in tight spaces and many field setups where the driving rows are differently spaced. The robot features a hybrid propulsion system: It is powered by a gasoline generator and batteries which allows environmentally friendly operation in green houses and also long outdoor runs. Bonirob is equipped with an onboard navigation system that enables the robot to navigate in fields [168]. The main operation mode is row-based navigation where the robot uses onboard sensors only (3D laserscanner, wheel odometry and an inertial measurement unit) to detect the field layout [169]. Additionally, external information like field maps or GPS/RTK GPS can be used to guide the robot in fields without detectable borders or to save information from the sensors with precise geo-referenced data.

The app implements the sensing, treatment or both functions. This modular design enables the reuse of the platform for different purposes throughout the year. Furthermore, this eases development of new robotic applications because only the app itself needs to be constructed, the generic navigation capabilities and robotic platform can be directly reused.

For plant classification and position estimation for single plant weed control a specific app is constructed for the Bonirob V2 as depicted in Figure 6.2.

The app comprises the newly developed camera system for plant classification and stem

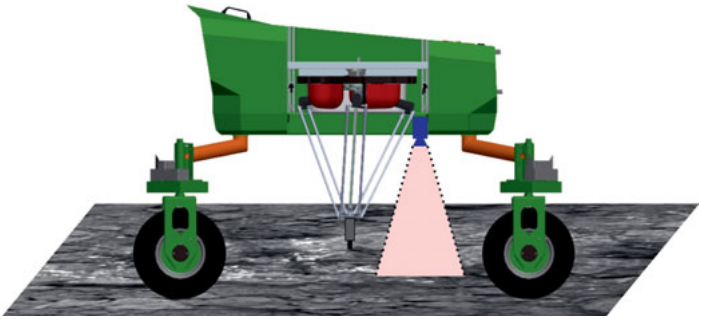


Figure 6.2: Simplified side view CAD drawing of the Bonirob V2 field robot for the Remote-Farming.1 use case: It features a camera for plant classification and position estimation as well as a delta robot with a mechanical weed treatment module.

detection as presented in Chapter 3 and contains a robotic manipulator for weed control [75]. The manipulator is a delta robot which is able to quickly move using a visual servoing camera to weed positions which are given by the plant classification and stem detection system (Chapters 4 and 5). At the tool center point of the delta robot a mechanical stamping tool is mounted. The tubular stamping tool [74] is able to push single weeds into the ground which is an effective method to regulate small weed plants in a mechanical and organical way. Possible alternative methods to treat single weed plants are for example use of small tines [170] or droplet based ultra precision spraying [171]. They are compatible with the robot and the developed plant classification and position estimation system.

Figure 6.3 shows in detail how the constructed app with vision system looks like. The Jai camera system is mounted together with an artificial illumination which lights up the viewing area of the Jai camera. Additionally, the black curtain around the app bay is visible which is applied to avoid effects from wind and direct sunlight. In the left part of the image the delta robot with the weeding tool can be seen.

An additional challenge arises because the Jai plant classification camera's field of view is different from the working area of the robotic manipulator and the field of view of the visual servoing camera. This is done on purpose: The robot arm and tool shall not block the Jai camera's field of view which allows the robot to operate continuously. All plant classifications and stem detections are associated to the time when the image is taken. Using the ego motion of the vehicle and the image timestamp, all plant positions are transformed into the robot end effector frame and refined using visual servoing such that the robot manipulator is able to position the weeding tool reliably at plant stem positions and keep it still while the weeding process is running.

The plant classification and stem detection system presented in this thesis is not restricted to be used with this field robot. Any platform (tractor, robot, drone, ...) and even hand-held image acquisition setups are possible. The field robot-based setup is used in the

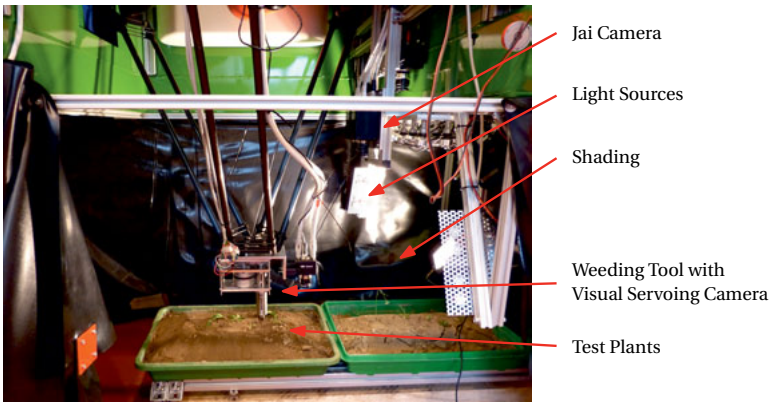


Figure 6.3: View into the app of Bonirob developed for organic weed control. The developed Jai camera system, shading and artificial light are visible.

experiments to achieve constant acquisition conditions that match the possible weed regulation process as closely as possible.

This Bonirob V2 field robot was used to both collect data for the evaluation of the plant classification and stem detection pipelines developed in this thesis and was used to demonstrate the whole system in a field trial. The robot was able to capture images, classify and detect the weed plants position and then to regulate the weeds using the tube stamp in experiments conducted within the publicly funded project RemoteFarming.1.

6.1.2 Recorded Field Image Datasets

For evaluation of the proposed plant classification and stem detection pipelines suitable training and test data is required. Therefore, the built Bonirob V2 field robot and the camera system developed in Chapter 3 were used to acquire field datasets in challenging outdoor organic farming conditions.

Unfortunately, no public datasets with annotations were available in the plant classification and stem detection domain. Public datasets play an important role for many tasks in computer vision [172, 173, 174, 175], machine learning [176] or robotics [177, 178, 179]. Some datasets were proposed in the agricultural domain for example in leaf recognition [180, 181, 182] and phenotyping of leaves of potted plants [183]. The lack of datasets for plant classification and/or stem detection in the field inhibits research. Every group works with their own data and comparison of results is difficult if not infeasible.

The goal of the dataset acquisition is to generate a real world dataset for plant classification and position estimation in the very challenging organic vegetable farming domain. For

crops like carrots, weed control must be carried out already for the first time in early growth stages where crop plants are smaller than 1 cm in diameter. Weed plants of any size can occur depending on when the farmer regulates weeds. Here, two datasets were acquired in different field locations and at different growing times.

Figure 6.4 displays the carrot field where dataset A was acquired. Carrots are grown on small dams to give the plants enough space to form nice carrots. The field shown here utilizes a single carrot row per dam growing system. Dataset B is acquired on a similar farm, however there two carrot rows are located next to each other on each dam.

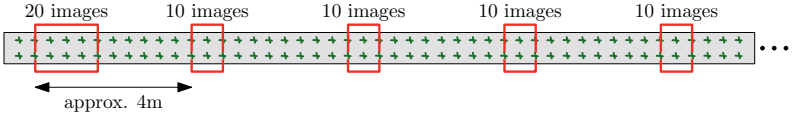
Figure 6.4: Overview image (left) of the carrot field where dataset A was acquired. Detailed view of one carrot dam with a lot of weed plants (right).

Field Layouts Figure 6.5 displays the field layouts of the two fields where the dataset recording took place. The images were captured in batches of 10 non-overlapping images each. The first batch contains 20 images to get more images from the start of the row so approaches which process data incrementally have a larger starting batch to process. The other batches are spread out along the field to capture different conditions which might occur. The difference in single crop row (dataset A) and dual crop row (dataset B) growing system can also be seen in the schematic field view.

Dataset Location and Recording Properties Table 6.1 summarizes the location, date and robot parameters where the datasets where acquired. Additionally, the average crop and weed sizes are given. These number are approximate because the diameter of crops and weeds varies throughout the field.

Dataset Overview and Challenges Figures 6.6 and 6.7 show example pictures of the two datasets. It can be observed that both crop and weed plants are much smaller in dataset A and that they are much more regularly sized. In dataset B most weed plants are very big compared to the crops, but also some small weed plants are present. The crop is planted in two rows.

(a) Field layout of dataset A



(b) Field layout of dataset B

Figure 6.5: Schematic visualization of field layouts for datasets A and B.

Table 6.1: Acquisition and field conditions of datasets A and B.

	Dataset A	**Dataset B**
Location	Gehrde, Germany	Lippstadt, Germany
Date	June 6, 2013	June 12, 2013
Camera	JAI AD-130GE	JAI AD-130GE
Robot	Bonirob v2	Bonirob v2
Driving Speed	4.5 cm/s	4.5 cm/s
Crop	Carrots	Carrots
Cultivation System	Single Row on Dam	Dual Row on Dam
Avg. Crop Diameter	1-2 cm	1-4 cm
Avg. Crop Spacing	2-3 cm	2-3 cm
Avg. Weed Diameter	1-6 cm	1-10 cm

The datasets contain challenging images with close-to-crop weed plants (for example in images A004 and B03) and lots of overlap between plants (for example in images A059, B003, B030 and B088). Additionally, the field situation changes along the row and images with few or many plants are present (compare for example images B018 and B030). Finally, the occurrence of plants in different sizes can be seen very well (in images of dataset B).

Plant Types Carrot plants in very early growth stages are the main crop to be classified. Figure 6.8a displays a collected set of carrot plants. First, the elongated seed leaves (first leaves that grow after germination, lat: cotyledon) are a distinct feature (see for example lower row, upper left carrot plant). The foliage leaves feature a distinct pattern and are pinnate shaped (they separate into smaller leaves in a tree-like structure).

Typical weeds encountered in the carrot fields are different species of broad leaf weeds; see Figure 6.8c. They feature roundish leaves and are less challenging to discriminate from

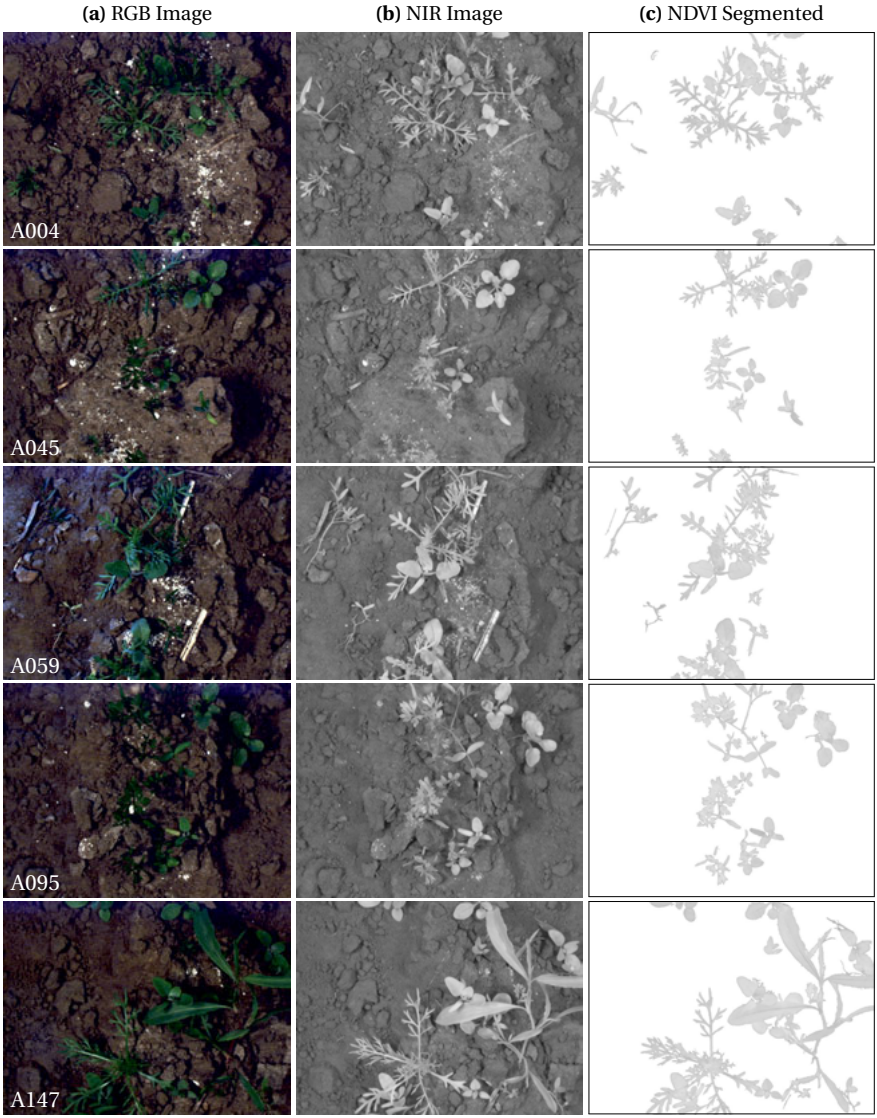


Figure 6.6: Sample images from dataset A. Depending on the location in the field many or few plants are present.

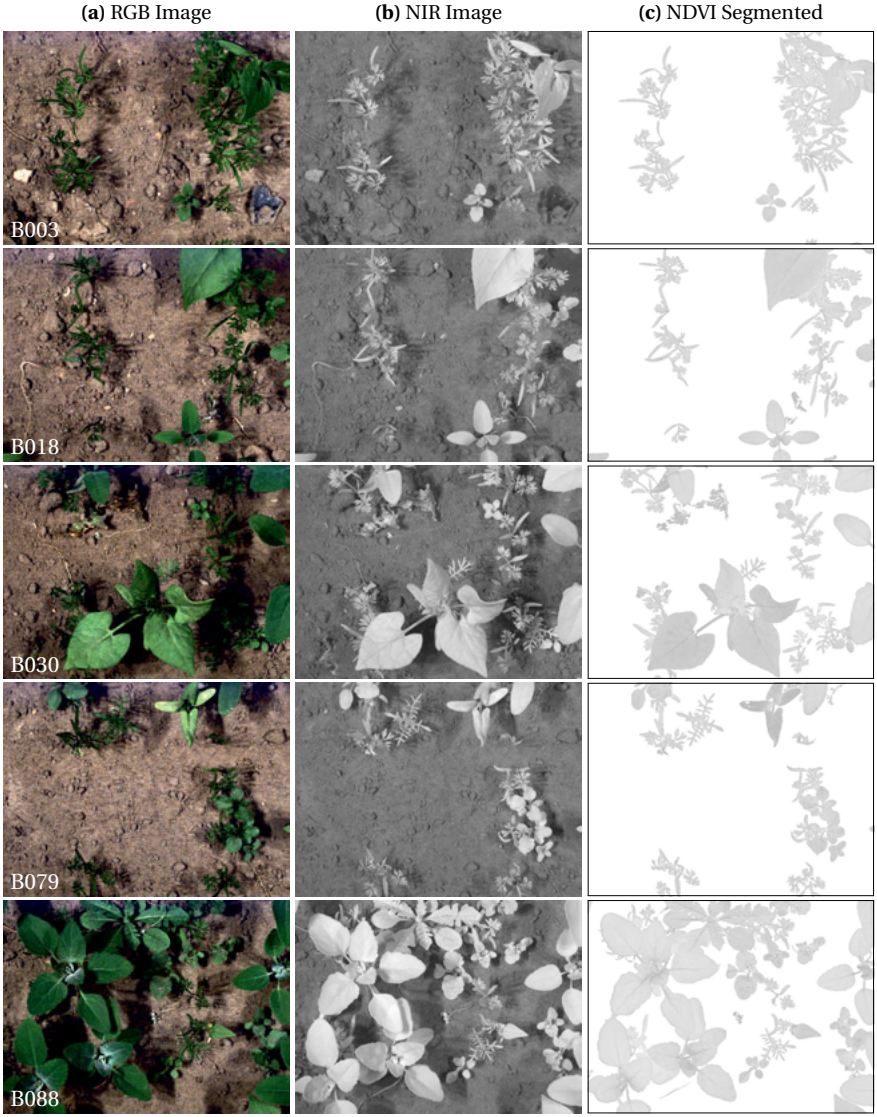


Figure 6.7: Sample images from dataset B. Compared to dataset A the weed plants are much larger while the crop plants span different sizes.

carrots compared to species like chamomile.

Chamomile plants are a very common weed with leaves which looks similar to carrots; see Figure 6.8b. The foliage leaves of both carrot and chamomile are pinnate shaped and difficult to discriminate. Therefore, all chamomile plants were labeled as a separated class. Without loss of generality the plant classification pipeline is tested with these three classes.

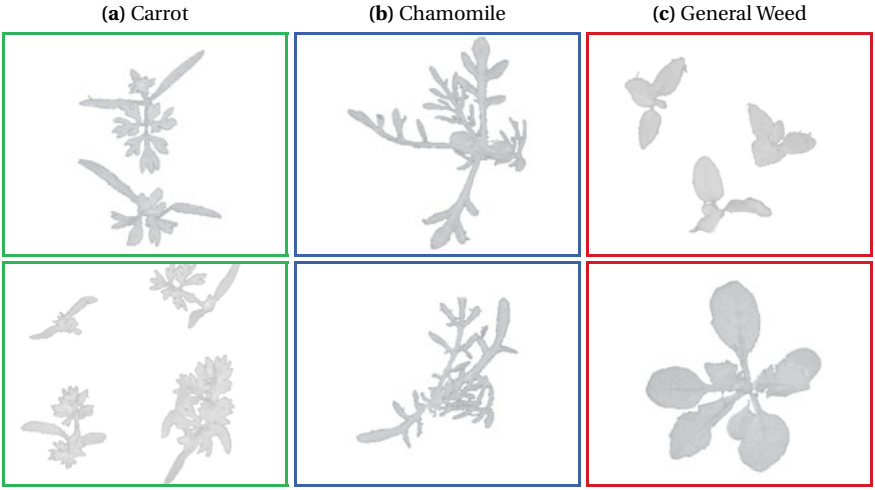


Figure 6.8: Sample segmented NDVI images for the three classes: Crop, chamomile and weed. All plants occur in smaller/larger variants in the datasets.

Ground Truth Labels The two field datasets are labeled with ground truth information: On the one hand, the labels consist of plant classification labels. Polygons are drawn to describe pixels belonging to the three different plant classes carrot, chamomile and general weed (see Section 4.3.1 for details). On the other hand, plant stem detection labels are added as points which mark centers of plants (see Section 5.3.1 for details). Table 6.2 summarizes the number of plant classification and plant position labels in the datasets.

Published Dataset For field-based plant classification to our knowledge no dataset was available to the public. Thus during the research for this thesis, a dataset with 60 images was recorded in the same field as dataset A and labeled with ground truth annotations (vegetation mask and crop/weed labels). It is called Crop Weed Field Image Dataset (CWFID) and is published online (`https://github.com/cwfid/dataset`). An accompanying paper [70] explains the dataset, discusses relevant image processing and classification tasks together with suitable evaluation metrics.

Table 6.2: Ground truth labels for the datasets A and B.

	Dataset A	Dataset B
Dataset Properties		
Number of Images	150	110
Growing System	Single Row	Dual Row
Plant Classification Labels		
Labeled Crop Regions	412	746
Labeled Weed Regions	613	799
Thereof Chamomile Regions	157	87
Thereof General Weed Regions	456	712
Total Labeled Plant Regions	1025	1545
Plant Position Labels		
Labeled Crop Stems	473	840
Labeled Weed Stems	698	790
Total Labeled Plant Stems	1171	1630

In the following, these two datasets are used to test and evaluate the developed plant classification and plant position estimation pipelines. Additionally, also results for the CWFID dataset are reported.

6.2 Evaluation and Discussion of the Plant Classification Method

In this section the plant classification system which is presented in Chapter 4 is evaluated and discussed. After a description of the evaluation method, the plant classification results are first studied visually and are then evaluated using suitable metrics. Finally, the plant classification results are discussed.

The evaluation of the plant classification pipeline applies leave one out cross-validation. Each image is selected as test image exactly once, all other images are used as training data to train a classifier for this test image. The test image passes through all processing steps of the pipeline and the output is a plant class label for each vegetation pixel in the image. The final metrics are calculated by summing all per-pixel results of all images and then by applying classification metrics (see Section 2.2.4).

One important aspect of the plant classification system is the choice of parameterization. In the following, the parameterization is chosen according to the parameter selection procedure introduced in Section 4.5 for dataset A. For dataset B the same methodology

is applied; the resulting plots are presented in Appendix A.2. Table 6.3 summarizes the chosen parameterizations for the two datasets A and B.

Table 6.3: Selected parameters for the plant classification pipeline when it is applied to datasets A and B. The parameters are determined as described in Section 4.5.

	Dataset A	**Dataset B**
General Parameters		
Patch Size w_{size}	80 px	60 px
Patch Stride w_{stride}	10 px	10 px
Smoothing Parameter		
Smoothing Parameter λ	1.2	0.6
Random Forest Parameters		
Number of Trees	100	100
Leaf Node Size	1	1
Number of Features Considered at each Split	4	4

The main difference in parametrization occurs for the patch size and the smoothing parameter λ. This can be explained by the different plant sizes as visualized in Figures 6.6 and 6.7: Dataset A contains a lot of chamomile plants which are significantly larger than the carrot plants, therefore the larger image patch helps to distinguish these from the crop and weed. The patch stride and Random Forest parameters do not differ for the two datasets. When going to a new field the parameterization can be chosen according how similar the situation is to dataset A or dataset B.

6.2.1 Results

Using the determined parameterization the plant classification pipeline is applied to the two datasets using leave one out cross-validation.

Visual Analysis of Plant Classification Output Figure 6.9 shows the input NIR image (column a), the color coded expert labeled ground (column b) and the color coded plant classification of the new pipeline (column c) side by side for images from dataset A. Figure 6.10 shows result images from dataset B. In addition each image is numbered; these numbers are used to refer to these images in the following analysis.

All in all, it can be observed that the system is able to classify the images into the three plant classes with high accuracy and uses all classes (crop, chamomile and weed). This works both for situations where only single plants are present and also in complex scenarios.

The system is able to distinguish overlapping plants and is not only returning a single plant class for large image blobs formed by grown together plants. This can be seen in images A059 and B003 where a lot of plants grow over and next to each other.

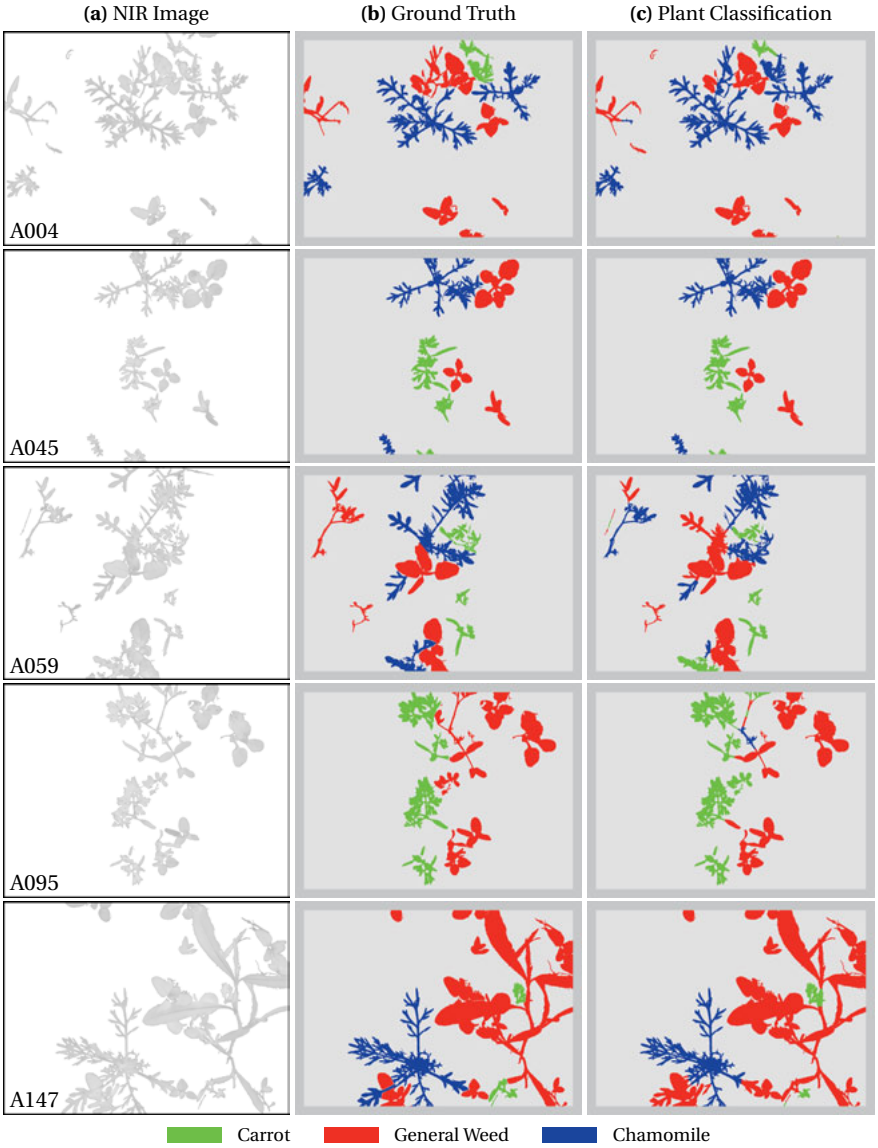


Figure 6.9: Plant classification results for dataset A.

Figure 6.10: Plant classification results for dataset B.

When analyzing the smoothness of the plant classification, it can be seen that the plant classification system is able to return consistent estimates for connected plant areas (for example images A004, A147 and B030): As desired, the plant class is not changing frequently between different estimated classes for adjacent pixels. This is especially visible for large plants for example in Figure 6.10 image B030.

The method is not perfect and some misclassifications occur: There exist few areas where the classifier misclassifies small parts of plants for example when a thin leaf crosses a large plant. Such a case can be seen in image A147 in the lower left part of the large chamomile plant.

Additionally, tiny long branch structures are occasionally classified into a wrong class because such branches occur for example both in the general weed and chamomile class and are therefore difficult to distinguish. In image A095 in the top middle a very elongated weed plant which crawls along the soil can be seen. In image B030 in the lower right a chamomile plant overlaps a crop and is misclassified as crop. The distinction between crop and chamomile is especially difficult for dataset B since only few chamomile plants are present in the whole dataset (see Table 6.2) and therefore training data for chamomile is limited.

Finally, the approach of not using a plant segmentation but the developed patch-based approach can split plants into multiple regions. Such a rare case happens in image B079. The chamomile plant in the top left area is split into a center and two outstretching areas. The center is misclassified as crop due to its appearance being very similar to a carrot. Such undesired splits in isolated plants only occur rarely and are heavily outweighed by situations where overlap occurs and is correctly treated by the pipeline.

Visual Analysis of Smoothing The smoothing step is a crucial component of the developed plant classification pipeline. Its impact can be seen in Figure 6.11, where in the left column (a) the classification results at each keypoint before smoothing and in the right column (b) after smoothing are plotted.

The smoothing process significantly improves the consistency of the plant classification output. Areas where the estimated plant class changes frequently are smoothed and large areas of the same plant class are achieved. This is in line with the assumption that plants are much larger than the stride of the patch extraction process and that a plant type has exactly one label. Using the smoothed plant classification image the interpolation method presented in Section 4.2.5 produces the full plant classification output which is also visualized in Figures 6.9 and 6.10.

6.2.2 Evaluation

The goal of this section is to go beyond visual inspection and to quantitatively evaluate the plant classification performance. First, an analysis using ROC curves is performed to analyze the classifiers ability to separate the different plant types. Second, classification

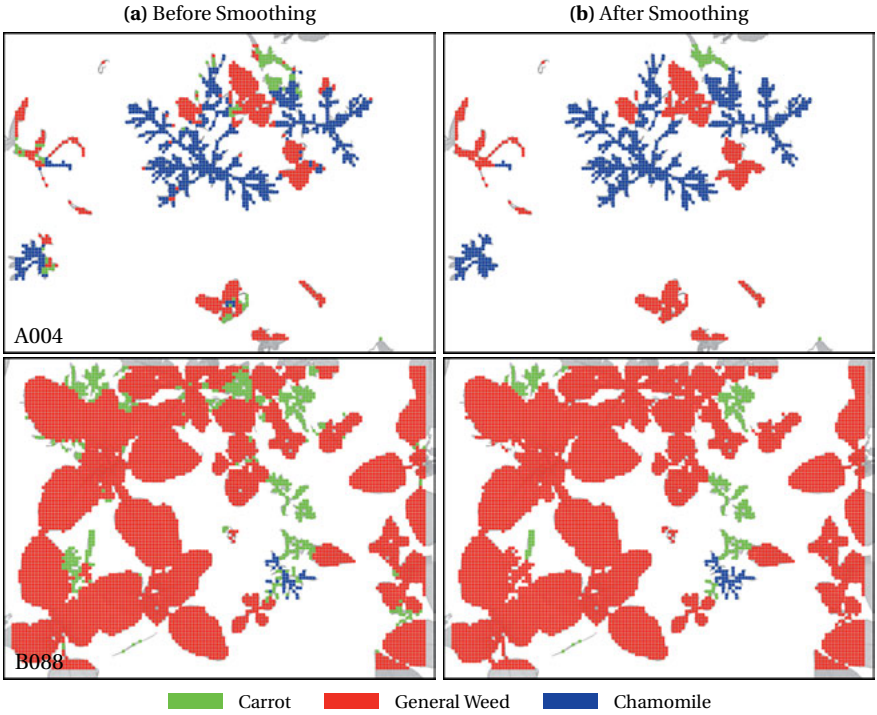


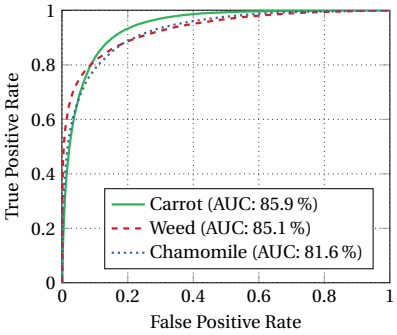
Figure 6.11: Plant classification label before (left) and after (right) smoothing plotted in color code at each keypoint for images from datasets A and B.

metrics according to Section 4.4 are applied to all vegetation pixels to quantify the effect of smoothing and to get final metrics for the overall plant classification task.

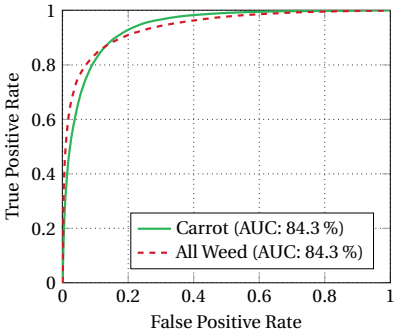
ROC Curve Analysis Using the score vectors $\boldsymbol{s}$, which is output by the Random Forest classifier before the smoothing step, a ROC curve is generated according to the definition and explanation in Section 2.2.4.

Figures 6.12a and 6.13a show ROC curves after leave one out cross-validation over all images for datasets A and B. In addition to the 3-class ROC curves, also a variant of the plant classification system with two classes is evaluated. Here, the labeled data is reduced to binary labels (crop vs. weed) and then the pipeline is run. These 2-class ROC curves are plotted on the right side in Figures 6.12b and 6.13b.

For dataset A the three ROC curves are well aligned and indicate equal classification performance for the three classes chamomile, crop and weed. When the dataset is reduced to a two class problem performance is roughly equal.

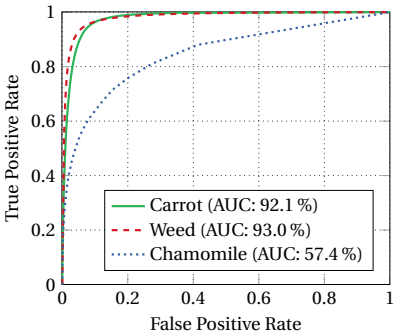


(a) 3-class ROC for Dataset A

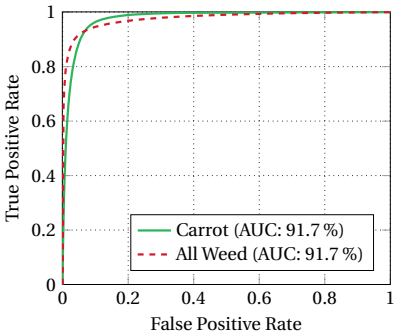


(b) 2-class ROC for Dataset A

Figure 6.12: One vs. all ROC curves for plant classification on dataset A. The left curve (a) shows the classification result for 3 classes, the right curve (b) shows the classification result for 2 classes where chamomile and weed are united into a single class called all weed. All results are before smoothing as described in the text.



(a) 3-class ROC for Dataset B



(b) 2-class ROC for Dataset B

Figure 6.13: One vs. all ROC curves for plant classification on dataset B. The left curve (a) shows the classification result for 3 classes, the right curve (b) shows the classification result for 2 classes where chamomile and weed are united into a single class called all weed. All results are before smoothing as described in the text.

When dataset B is analyzed, it can be observed that the distinction of chamomile from the other classes is difficult. Causes for this are: First, the dataset B contains a lot less data for chamomile than for crop and weed. Furthermore, the chamomile plants are thin and pinnate shaped and therefore do not produce many patches. Second, the chamomile plants in dataset B are relatively small (compared to dataset A) and are of similar size as the crop which makes distinction between the two difficult (even for a human). When dataset B is reduced to a two class problem, the ROC curves for crop and weed both improve to very high levels.

The ROC analysis is performed on unsmoothed data since classifications scores are required. The smoothing process increases the performance additionally and is analyzed in the following.

Smoothing The smoothing procedure is a crucial step in the plant classification pipeline. It improves performance by introducing spatial smoothness as shown in Figure 6.11 above. Here, the influence of smoothing on the whole dataset is analyzed and quantified.

Once the smoothing step is applied, the plant classification scores $\boldsymbol{s}$ are transformed into smoothed plant class labels $\hat{l}$. For these only a single point of a ROC curve could be plotted and another type of evaluation is conducted: The average accuracy, precision, recall and F1-score are determined before and after smoothing for all biomass pixels in each image. Table 6.4 shows these plant classification performance metrics before and after smoothing on the datasets A and B.

Table 6.4: Improvement through smoothing on datasets A and B.

	Accuracy	**Precision**	**Recall**	**F1-score**
Dataset A				
No Smoothing	87.3 %	78.8 %	78.3 %	78.5 %
After Smoothing	91.4 %	86.2 %	84.4 %	85.3 %
Improvement	+4.1 pp	+7.4 pp	+6.1 pp	+6.7 pp
Dataset B				
No Smoothing	95.4 %	81.1 %	66.8 %	73.2 %
After Smoothing	96.7 %	93.6 %	66.3 %	77.6 %
Improvement	+1.4 pp	+12.6 pp	-0.5 pp	+4.4 pp
Average Improvement	+2.8 pp	+10.0 pp	+2.8 pp	+5.6 pp

The quantitative evaluation clearly proves, that smoothing improves the plant classification results in both datasets.

Additionally, a benefit beyond the pure improvement in classification metrics is, that the smoothing step makes the estimate more homogeneous as it inhibits misclassifications for single patches in or around a large homogeneously classified area. This is beneficial when the plant classification output is used for example for weed control.

Classification Metrics Finally, the overall plant classification performance of the developed pipeline is analyzed with metrics. It shall be noted again that all metric calculations are performed for vegetation pixels only and background is ignored (see Section 4.4 for all details). Table 6.5 summarizes these metrics for each class and in total for the datasets A and B.

Table 6.5: Final plant classification results for datasets A and B measured with classification metrics average accuracy, precision, recall and F1-score after smoothing.

(a) Plant classification results after smoothing for dataset A.

Dataset A	Average Accuracy	Precision	Recall	F1-score
Per Class				
Carrot	91.8 %	82.8 %	87.3 %	85.0 %
Chamomile	92.4 %	86.2 %	73.8 %	79.5 %
Other Weed	90.0 %	89.6 %	92.0 %	90.8 %
Overall	91.4 %	86.2 %	84.4 %	85.3 %

(b) Plant classification results after smoothing for dataset B.

Dataset B	Average Accuracy	Precision	Recall	F1-score
Per Class				
Carrot	95.6 %	89.4 %	91.3 %	90.4 %
Chamomile	97.9 %	94.7 %	8.6 %	15.7 %
Other Weed	96.7 %	96.8 %	98.9 %	97.8 %
Overall	96.7 %	93.6 %	66.3 %	77.6 %

On both datasets high average accuracies of 91.4 % and 96.7 % are achieved which proves the performance of the plant classification system quantitatively. The precision values (which rate how correct the assignment to a class is) of 86.2 % and 93.6 % show that the system is able to produce accurate estimates.

The recall, which defines how many instances of a class which are present in the training data were correctly classified, is in the same range as the precision for dataset A. For dataset B the per class recall for the chamomile class is low. Here the same explanation as given above applies; the dataset contains few training examples and the chamomile plants look very similar to the crop plants. Note that also for dataset B the overall recall is 66.3 % and that the individual per class recalls for crop and weed classes are above 90 %. The F1-score is calculated from precision and recall as defined in Section 2.2.4 and supports the previous judgment.

All in all, the evaluation of the plant classification system on the two datasets A and B using metrics yields the same conclusions as visual inspection.

Cross-Evaluation So far the evaluation is performed on the two datasets separately. Now, a cross-evaluation is performed where the pipeline is trained on one dataset and then applied to the other. This simulates the use case where a farmer trains the plant classification system once and then applies it to different fields.

Table 6.6 presents the cross-evaluation results for the plant classification pipeline. The first row of each subtable displays the plant classification metrics when parameterization is done on one dataset and the pipeline is applied to images from the other dataset. In both cases (dataset B → dataset A and dataset A → dataset B) the results are lower compared to the case when the pipeline is trained and applied to the same dataset.

Table 6.6: Cross-evaluation plant classification results after smoothing. In subtable (a) the first row A → B presents results of the pipeline parameterized and trained on dataset A applied to dataset B. In the second row A+10 → B^{-} results for parameterization on dataset A and training on dataset A plus the first 10 images of dataset B applied to remainder of dataset B are given. For the rows in subtable (b) the same methodology applies.

(a) Cross-evaluation dataset B → dataset A.

	Average Accuracy	**Precision**	**Recall**	**F1-score**
B → A	79.1 %	69.4 %	57.1 %	62.6 %
B+10 → A^{-}	88.2 %	77.8 %	80.4 %	79.1 %

(b) Cross-evaluation dataset A → dataset B.

	Average Accuracy	**Precision**	**Recall**	**F1-score**
A → B	93.4 %	71.2 %	68.4 %	69.8 %
A+10 → B^{-}	96.1 %	79.8 %	62.8 %	70.3 %

The second row in each subtable Table 6.6a and 6.6b presents a case where 10 additional images from the target dataset are labeled. In the real world this is the use case where the farmer spends a short time to label for example 10 images of the target field and performs a retraining (one click) before applying the plant classification system. These additional 10 images allow the plant classification system to perform substantially better in both cross-evaluation cases. Compared to Table 6.5 only a slight drop in performance occurs. Average accuracy drops only slightly by 3.2 pp or 0.6 pp. The F1-score drops 6.2 pp or 7.3 pp but stays at a good level above 70 % for both cross-evaluations.

Crop Weed Field Image Dataset Finally, the pipeline is applied to the Crop Weed Field Image Dataset (CWFID) dataset published in conjunction with [70]. The results are summarized in Appendix A.1 and agree with the results achieved for datasets A and B.

6.2.3 Discussion

The contribution is a new plant classification system which is applied to the task of crop/weed discrimination in commercial crop fields. The system processes multispectral images from a camera and does not require additional input. It can be applied on a moving field robot like the Bonirob introduced above and processes images in real time.

Two datasets from two different organic carrot farms are used to evaluate the system in the previous section. The evaluation comes to the conclusion that the plant classification pipeline is able to process these datasets with high accuracy: 91.4 % and 96.7 % are achieved for dataset A and dataset B respectively.

The system successfully resolves complex situations present in the dataset: For example dataset A contains very small plants and difficult areas where many plants grow close together. The second field dataset B features a much larger variety of plant size (especially very large weed patches) and overlap between all plant types is present. Additionally, the weed chamomile – which is labeled as separate class – poses an additional challenge: Chamomile looks similar to carrot plants and is difficult to discriminate.

Both of the two main approaches in related work do not handle these challenges present in fields with crop in early growth stage well:

First, plant- and leaf-based methods struggle when plants overlap. These methods apply prior plant/leaf segmentation which is error-prone for overlapping plants and an unsolved challenge. This results in the classification performance to significantly decrease when overlap is present as discussed in Section 4.1.

Second, cell-based methods suffer from reduced output resolution. They only estimate a single plant class label for an entire cell spanning hundreds of pixels and potentially multiple plants. This is not desired and not enough to solve the task of plant classification for single plant weed control.

The newly developed plant classification pipeline overcomes these limitations because it neither requires a plant/leaf segmentation nor does it output only coarse per cell classification results. Complex situations like close-to-crop weeds and overlap between plants (especially intra-class overlap) are successfully tackled by applying the patch-based feature extraction and classification steps. Since no prior leaf or plant segmentation is performed, segmentation errors cannot influence the classification result. The system generically handles field situations with and without overlap and there is no special functionality to cope with overlap.

The novel smoothing scheme is an additional key contribution. It ensures smooth classification results and avoids rapidly changing plant class estimates.

In order to not limit the plant classification output to a per patch result, the interpolation step ensures the generation of a full plant classification output image. For each vegetation pixel a plant class is estimated. This is a significant improvement over cell-based methods presented in related work.

The evaluation shows that overlap between different classes of plants is detected with high accuracy. One limitation is however, that multiple overlapping plants of the same class (intra-class overlap) are not separated into different plant regions. In the output label image they get represented by one connected component of the same plant class.

This is no drawback for the goal of the thesis. The targeted application is precision weed control and in that use case overlap between different classes (inter-class overlap) is required to be discriminated properly. Precision agriculture metrics (for example weed coverage) are not influenced since they operate on the individual pixels in the classification image.

The cross-evaluation proves that the plant classification system is also able to generalize beyond the specific dataset used to train the system. This is a major strength of the presented approach since it allows practical application by farmers: Full relabeling, parameterization and retraining of the system for a new field is not required and the farmer can apply an existing classifier on a new field. If a small amount of labeling in the new field is acceptable, the results can be further improved. The drop compared to full training is limited to a maximum of 7.3 pp F1-score and 3.2 pp average accuracy when cross-evaluating with datasets A and B. Such additional labeling can be for example performed in field with a smartphone application or if the application is data acquisition for phenotyping such additional labeling can also be performed offline as postprocessing step.

The output of the system is a plant classification image that can be used for precision weed control with a field robot. Additionally, the pixel labels can be used to calculate weed coverage or the crop/weed area ratio metrics that help farmers when applying precision agriculture techniques on their fields.

All evaluations and results are presented for the more challenging three class task. The special weed chamomile which looks similar to the carrot crop is labeled as separate third class. The goal is to show that this pipeline is more versatile than pure crop/weed discrimination and still returns high quality plant classification images. For pure crop/weed classification a two class classification would be sufficient and the ROC analysis indicates that using the pipeline with two classes results in even better classification accuracies.

Plant classification was defined as objective for this thesis and the developed pipeline fulfills the defined requirements: It is able to process small plants in early growth stage (0 cm to 5 cm in diameter), works with high resolution field capable sensors and as derived in the evaluation copes with overlap and challenging field conditions.

Moreover, the following two objectives are fulfilled as well: The plant classification system performs without human supervision when it processes new images and runs in real time on a CPU. The plant classification system generalizes beyond the concrete dataset it is trained on and can be applied to similar data from other fields or seasons. All in all, this allows plant classification with a field robot and enables automatic organic weed control when combined with a proper weeding tool.

6.3 Evaluation and Discussion of the Plant Position Estimation Method

The goal of this section is to apply the new plant position estimation pipeline developed in Chapter 5 to challenging real world datasets, to evaluate and to discuss these results. To generate plant position estimates, the pipeline is applied to the two datasets A and B using 5-fold cross-validation.

An important aspect of the plant position estimation system is the choice of parameterization. The methodology presented in Section 5.5 is applied to determine the best parameterization for each dataset. The parameter selection plots for dataset A are presented in Section 5.5 and for dataset B they can be found in Appendix A.3. Table 6.7 summarizes these parameterizations for the datasets A and B.

Table 6.7: Selected parameters for plant position estimation on datasets A and B.

	Dataset A	**Dataset B**
General Parameters		
Patch Size w_{size}	40 px	50 px
Patch Stride w_{stride}	10 px	8 px
Training Patch Extraction Parameters		
Border around Ground Truth Stems d_{border}	35 px	40 px
Maximum Chebyshev Distance to Stem d_{max}	12 px	9 px
Sampling Step d_{step}	2 px	2 px
Position Estimation Parameters		
Smoothing Kernel Size k_{smooth}	3	3
Non-maximum Suppression Kernel Size $k_{\text{non\_max}}$	15	15
Random Forest Parameters		
Number of Trees	120	120
Leaf Node Size	1	1
Number of Features Considered at each Split	3	3

The optimized parameterizations of the plant position estimation pipeline only differ slightly between the two datasets. The patch size is 10 px larger for dataset B which can be explained with the larger plant size in dataset B. A larger patch size is beneficial to capture the appearance of the stem region of the larger plants. The training patch extraction parameters are nearly the same and the difference in d_{max} has minimal effect on performance as shown in the respective plots in Figures 5.18 and A.7. The border parameter d_{border} is roughly proportional to the patch size which makes sense to ensure stem and non stem patches do not overlap. The position estimation and Random Forest parameters do not depend on the dataset and are equal for both datasets A and B.

Interestingly, when comparing the patch size for plant classification and position estimation, dataset B requires a smaller patch size for plant classification and a larger patch size for position estimation. This is not a contradiction and an explanation to this could be to larger size and bigger homogeneity of plant appearance present in dataset B compared to dataset A. Larger plants require less context for plant classification compared to the many pinnate chamomile plants present in dataset A. The larger plants in dataset B also have a bigger stem region for which more context helps precise position estimation.

One additional position estimation parameter motivated by the organic farming use case is the stem detection threshold γ. It determines how far the detected position can deviate from the ground truth position to be still considered valid. The organic weed control process presented above is the considered use case for the pipeline. The tube stamp weeding tool developed (see Section 6.1.1) is now considered to define the stem detection threshold γ. The diameter of the tool is 10 mm and it is considered efficient 1.5 times its radius. Together with the average pixel resolution of 8.95 px/mm (see Table 3.2) at nominal object distance, the stem detection threshold γ is set to 68 px. In metric scale this equals 7.5 mm.

6.3.1 Results

The novel plant position estimation pipeline is now applied to all images in a dataset using 5-fold cross-validation and the defined parameterization. After processing all images, detected plant stem positions are available for each image. They can now be analyzed visually and compared with the ground truth stem positions.

Figures 6.14 and 6.15 display the plant stem position estimation results for datasets A and B respectively. The masked NIR image is augmented with detected and ground truth stem positions.

Green stars visualize the detected stem positions which are the output of the final non-maximum suppression step. Additionally, the ground truth stem positions are visualized while taking into account the following two cases:

1. Ground truth stem position to which a detected position is associated are visualized with a blue plus. These correctly detected ground truth stem position meet the stem detection threshold γ from Equation (5.2). In addition to the blue plus indicating a correctly detected ground truth position, a red line connects this ground truth position to the corresponding detected stem position.
2. Missed ground truth stem positions are visualized with a red plus. For these no detected stem position which satisfies the stem detection threshold γ could be associated. Therefore, these ground truth positions are not connected to a detection with a line.

The set of false alarms, i.e. additional invalid detections, can be identified because these green stars do not have a red line connecting them to a ground truth position.

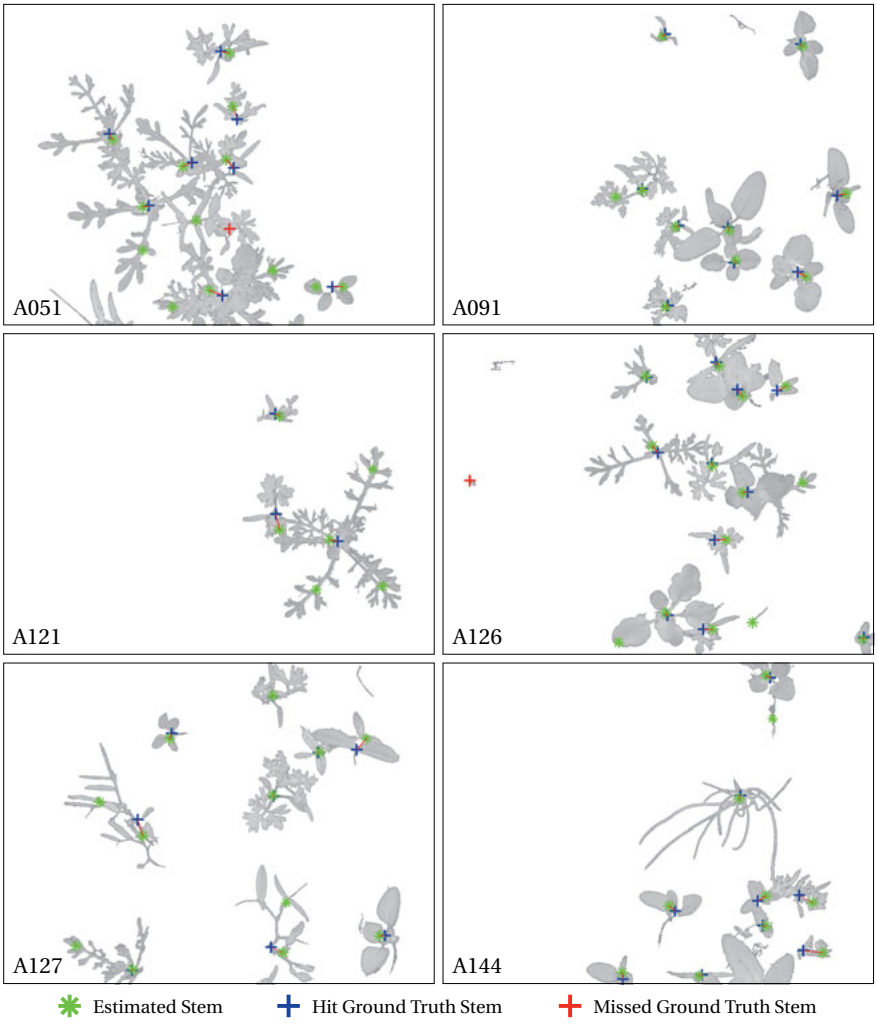


Figure 6.14: Plant position estimation results for dataset A. Green stars mark detected plant positions. Blue pluses mark ground truth stem positions for which a detection is present. Red pluses mark ground truth stem positions which were missed (and for which no detected stem is present).

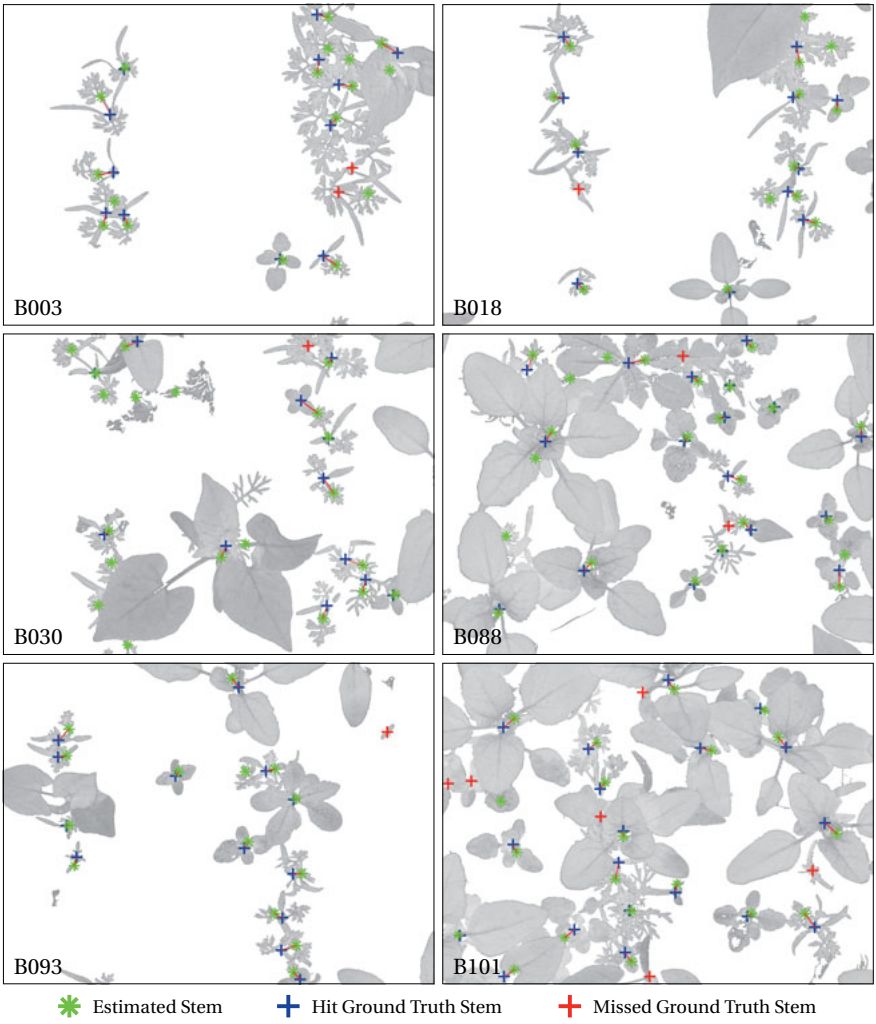


Figure 6.15: Plant position estimation results for dataset B. The symbols are explained in the legend below the image and in the caption of Figure 6.14 in detail.

When the images are analyzed the difficulty of the plant stem detection task can be observed. The plant stem region varies a lot even for carrot plants. The majority of stems is detected both in datasets A and B. Misses happen more often in dataset B than in dataset A. One reason for this is the greater density and variance in size of plants in dataset B. The larger a plant, the more leaves are present in different configurations close to the stem compared to early growth stages.

The visual analysis suggests that in both datasets false alarms happen more often than misses. For example in image A121 it can be observed that along the large leaves of the chamomile plant (lower right) multiple false alarms happen. The area where tiny leaf parts branch of left and right of the main pinnate leaf look very similar to the carrot plant stem area. Additionally, when overlap is present additional false alarms are trigged: For example in image A051 in the lower part of the image lots of plants grow together. Such overlap does however not always produce false alarms, the three plants growing in A126 close to the top right corner overlap but their stems are correctly detected without additional false alarms.

From the images of dataset B it can be concluded that the stem detection pipeline copes well with plants of different size. Especially, in image B030 and B088 plants of very different growth stage are present and correctly handled.

All in all, the plant position estimation pipeline is able to estimate the stem region correctly for the majority of plants. Misses happen occasionally but more false alarms are present. For the mechanical weeding process false alarms are not as harmful as misses. The false alarms trigger an additional weed removal action; if this action is fast this does not degrade the weeding performance, it only might slow down the overall process.

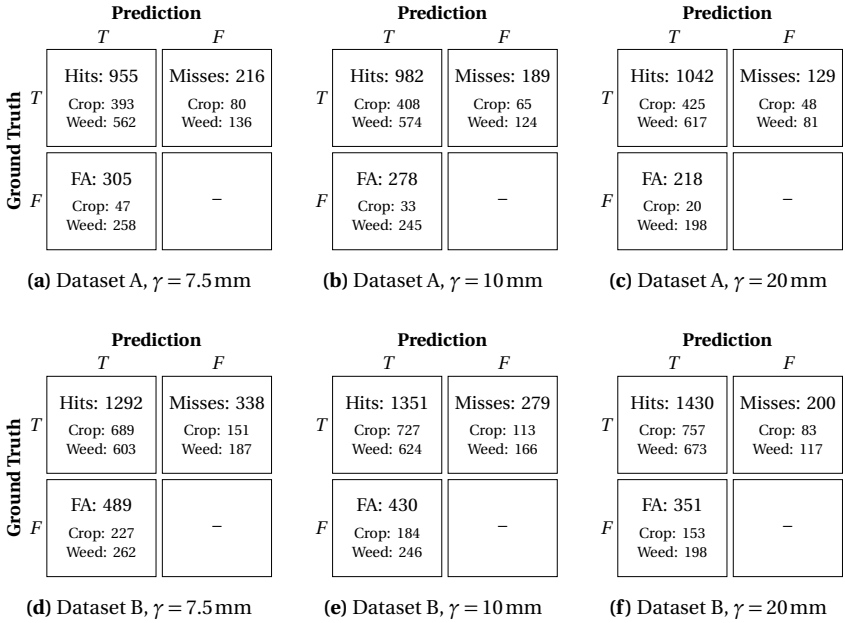
6.3.2 Evaluation

The previous section presents the plant position estimation pipeline results visually. Now, a quantitative evaluation using the plant stem detection evaluation metrics and ground truth annotations follows.

Throughout the evaluation, detected and ground truth plant stem positions are categorized into hits, misses and false alarms using the stem detection threshold γ. The threshold γ describes how far an estimate is allowed to deviate from the ground truth to still be counted as valid according to Equation (5.2).

Plant Position Estimation Confusion Matrix Using the hits, misses and false alarms the plant position performance can be evaluated with a confusion matrix. In Figure 5.12 the confusion matrices are plotted for the three thresholds 7.5 mm, 10 mm and 20 mm for datasets A and B.

The 7.5 mm threshold is the main threshold motivated by the agricultural process of mechanical weed control process described above in Section 6.1.1, the other thresholds allow more deviation and are useful for plant counting or other precision agriculture



(a) Dataset A, $\gamma = 7.5\,\text{mm}$ **(b)** Dataset A, $\gamma = 10\,\text{mm}$ **(c)** Dataset A, $\gamma = 20\,\text{mm}$

(d) Dataset B, $\gamma = 7.5\,\text{mm}$ **(e)** Dataset B, $\gamma = 10\,\text{mm}$ **(f)** Dataset B, $\gamma = 20\,\text{mm}$

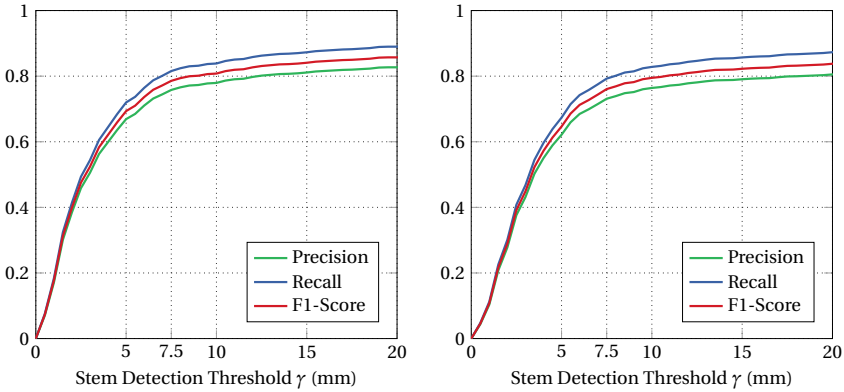
Figure 6.16: Plant position confusion matrices for datasets A and B with different plant position estimation thresholds γ. FA abbreviates false alarms.

processes. The confusion matrix analysis supports the previous statement that more false alarms (FN) are present than misses (FP). The majority of plant stem regions are correctly detected as hit (TP) for both datasets A and B. Additionally, the data in the confusion matrices is given for the crop and weed classes. It can be observed that false alarms happen more often for the weed class.

Plant Position Estimation Plot Instead of a confusion matrix, the hit, miss and false alarm counts can be used to calculate the plant position estimation metrics precision, recall and F1-score (see Section 5.4). Figure 6.17 plots these metrics for different plant stem detection thresholds γ from 0 to 20 mm.

From the figures it is clear that as expected with growing threshold all performance metrics grow. For thresholds between 0 and 5 mm performance grows fast, above 15 mm the performance grows slower and begins to saturate.

Plant Position Estimation Metrics In addition to plotting, Table 6.8 gives the plant positions estimation metrics in percent for selected thresholds starting with 7.5 mm. The plant position estimation pipeline achieves F1-scores for datasets A and B of 78.6 % and 75.8 % respectively for the selected threshold γ of 7.5 mm. For larger thresholds all metrics



(a) Metrics for Dataset A.

(b) Metrics for Dataset B.

Figure 6.17: Plant stem detection metrics precision, recall and F1-score plotted for varying values of the threshold γ for datasets A and B. As expected, the performance grows for larger thresholds.

improve for all datasets.

From the table it becomes clear that both precision and recall are approximately equal. This is achieved because maximization of the F1-score is the objective in the parameter selection procedure explained in Section 5.5. If on the one hand a higher recall is desired, parameters can be selected such that the hit vs. miss ratio improves. If on the other hand, false alarms should be avoided, parameters which maximize recall can be selected. Since this depends highly on the application, the balanced case is chosen here and good results for precision and recall are achieved.

Similar to the evaluation of the plant classification pipeline also for the plant position estimation a cross-evaluation is performed: The pipeline is parameterized and trained on dataset A and then applied to dataset B which results in a drop in F1-score of 9.8 pp. For the reverse case (dataset B → dataset A) a drop in F1-score of 7.7 pp is measured. The drop can be explained by the large difference in plant stem appearance in the two datasets. Defining additional plant position ground truth is a much easier task (a single mouse click) compared to labeling plant contours in the plant classification case. For full performance additional labeling of plant centers can be performed when the pipeline is applied to new fields with novel plant appearance.

6.3.3 Discussion

The novel plant stem detection and position estimation pipeline is a contribution towards precision agriculture and robotic mechanical weed control in commercial crop fields.

Table 6.8: Evaluation of the plant position estimation pipeline with metrics for datasets A and B. The metrics are evaluated for different thresholds γ; the 7.5 mm threshold for the carrot use case is printed in bold.

(a) Plant position estimation results for Dataset A.

Threshold γ	Average Accuracy	Precision	Recall	F1-score
$\gamma = 7.5$ mm	64.7 %	75.8 %	81.6 %	78.6 %
$\gamma = 10$ mm	67.8 %	77.9 %	83.9 %	80.8 %
$\gamma = 15$ mm	72.5 %	81.1 %	87.3 %	84.1 %
$\gamma = 20$ mm	75.0 %	82.7 %	89.0 %	85.7 %

(b) Plant position estimation results for Dataset B.

Threshold γ	Average Accuracy	Precision	Recall	F1-score
$\gamma = 7.5$ mm	61.0 %	72.5 %	79.3 %	75.8 %
$\gamma = 10$ mm	65.6 %	75.9 %	82.9 %	79.2 %
$\gamma = 15$ mm	69.6 %	78.6 %	85.9 %	82.1 %
$\gamma = 20$ mm	72.2 %	80.3 %	87.7 %	83.8 %

Using vegetation segmented multispectral images as only input, the pixel positions of plant stems in the image are estimated. The output is a list of estimated plant stems for each image. The system runs on a CPU in real time and can be applied on field robots.

Using the two challenging datasets A and B recorded in commercial carrot farms the system is evaluated. The system achieves a stem detection F1-score of 78.6 % and 75.8 % respectively which is well suited for weed control with the current mechanical tool. The selected threshold of 7.5 mm is very strict; it allows weed control in organic farming with a mechanical weed control tool. If the larger tool and therefore larger threshold of 20 mm is selected performance improves to 85.7 % and 83.8 % respectively.

Compared to related work the system has three clear advantages: First, the presented pipeline does not require a plant or leaf segmentation. Related methods perform such a segmentation as preprocessing step and then derive the plant stem using centroid or related methods. As previously discussed plant/leaf segmentation is error-prone in the field situation encountered here with overlap and a vast variety of plant sizes. Second, the pipeline is able to estimate the plant stem of all plants in the field. Related methods which apply crop row extraction and use the row estimate as input are only able to determine the crop positions. Similarly, crop mapping based approaches map the crop position and are therefore only able to report crop positions. Third, the approach only utilizes images and does not rely on additional sensing methodologies like GPS or 3D data.

The evaluation with plant position estimation metrics proves the high performance of the system. However, some of the labeled stems are missed. Misses happen for example when overlap is present or when plant stem predictions are too close together and the non-maximum suppression rejects one of the stems. Furthermore, an error case is that additional false alarms happen. This can occur for example for chamomile and carrot plants: Both species have pinnate leaves and some leaf areas (especially where small leaf parts branch of the main leaf) look similar to plant stem regions. Also, overlap between plants can create patches that look similar to stems in the top-down view.

The cross-evaluation indicates that the plant position estimation pipeline generalizes beyond the dataset, however a drop of performance occurs. This drop is between 7.7 pp and 9.8 pp in the F1-score metric depending on the datasets. Additional labeling is easy and possible in real time since defining a position estimate is a single click on the computer or touch on a smartphone. The farmer can decide based on the field situation (current vs. training field) if the pipeline is applied as it is, or if he wants to define additional labels for maximum performance.

The newly introduced pipeline overcomes the limitations of related work by design and is able to output plant position estimates for all plants from images only. The plant stem position estimation system handles complex field images with plants of different types, sizes and with intra- and inter-class overlap. The system is trained with stem labels only and predicts stems for all plant types. There is no need to train the system for specific plant classes separately.

The evaluation in challenging field conditions with data recorded with a field robot indicates the applicability of the pipeline to the mechanical weed control task. Furthermore, the plant position estimates can be utilized for additional precision agriculture tasks such as plant counting, plant mapping, single plant fertilization and pruning.

The presented plant position estimation pipeline fulfills the objective of this thesis to determine the position of a plant in the field for plants in early growth stage. The performance is shown for so far unsolved challenging real world carrot field images with plants which vary in size (both crops and weeds can be tiny or large) and with intra- and inter-class overlap being present. Additionally, the objective to work automatically in the field without human input and the ability to work in field robots in real time is accomplished.

6.4 Combined System for Weed Control

The plant classification and position estimation pipelines can be combined to realize single plant weed control. Therefore, the image is processed in parallel with the two pipelines and the results are combined to calculate weed treatment positions. An overview of this approach is displayed in Figure 6.18.

Since plant position estimates are calculated for all plants in the image (both crop and weed), they need to be combined with the estimated plant classification. This is done by

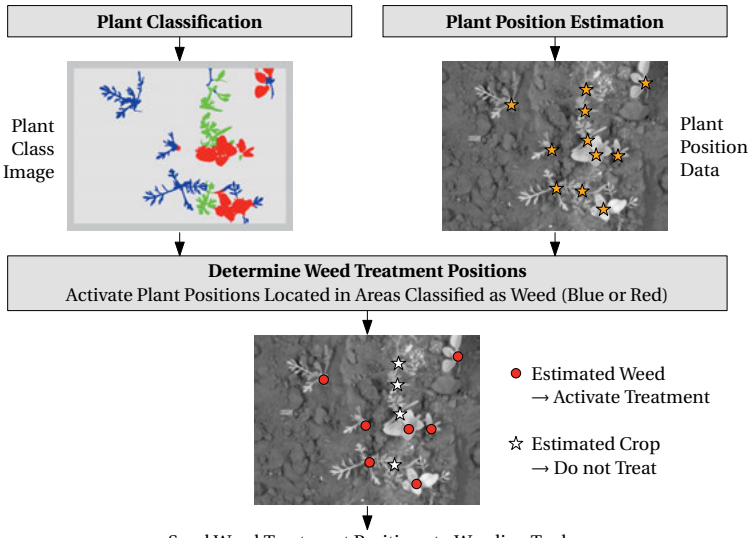


Figure 6.18: Combination of the plant classification and position estimation systems to realize single plant weed control.

looking up the plant class of each stem detection in the plant classification image. If a plant position belongs to a weed class, it is marked as active and the position is send to the weed control module such that the weed is treated. This can be seen in the lower part of Figure 6.18. If the plant position belongs to the crop class, the position is not activated and no treatment is performed.

To evaluate the combined system from a farmer's perspective the following approach is taken:

1. The farmer desires to treat as much weed as possible: For each ground truth weed plant position it is now evaluated (with the stem detection threshold γ) whether the weed plant was detected and activated or not. Ground truth positions for which an activated plant position is associated are counted as successfully treated. This number is now compared to the total amount of weed plants in the dataset to define the fraction of successfully treated weed plants.
2. Crop plants shall be protected and not treated: The same calculation as for weed is done and the amount of wrongly treated crop plants is calculated using the position estimation threshold γ and the activated positions associated to a crop ground truth.

This task is challenging since to achieve successful weed treatment both the plant classification must be successful (detect the correct plant class for the plant) and additionally the

plant position estimation must detect a plant stem close to the ground truth labeled stem (since the stem detection threshold γ is only 7.5 mm). Failure of either pipeline results in untreated weed or loss of crop which both are not desirable.

False alarms, which are additional plant position estimates not satisfying the plant position threshold γ, do not affect these metrics since the weeding tool is only effective if it reaches the plant stem. Such additional treatments only slow down the weeding process with additional weeding tool executions.

Table 6.9 displays the results of the combined system on the datasets A and B as well as the sum over both datasets.

Table 6.9: Evaluation of the combined system for stem threshold $\gamma = 7.5$ mm.

	Treatment Count	**Treatment Percent**
Dataset A		
Weed Treated	559 of 698	80.1 %
Crop Treated	79 of 473	16.7 %
Dataset B		
Weed Treated	570 of 790	72.2 %
Crop Treated	49 of 840	5.8 %
Sum of Datasets A and B		
Weed Treated	1129 of 1488	75.8 %
Crop Treated	128 of 1313	9.7 %

It can be concluded that for datasets A and B the combined system successfully treats 75.8 % of weeds and accidentally removes 9.7 % of crop plants. In dataset A where many chamomile plants occur and the distinction between crop and chamomile is difficult the number of lost crop is higher; at the same time the ratio of treated weed is also higher. In dataset B crop and weed is well distinguished and only 5.8 % of crop is lost. However, due to the high variance in weed size and appearance as well as much overlap, more weed plants are missed.

This combined evaluation of the plant position estimation and classification pipeline concludes the experiments and presents results from a farmer's perspective: A farmer is able to utilize the system to treat weed effectively with only minimal loss of crop. Additionally, the overall performance on crop yield is further increased beyond the presented numbers: Typically multiple weeding runs are performed in the first weeks after germination and additionally more crop than necessary is sown to compensate for loss of crop due to natural and weeding effects.

6.5 Summary

This chapter presents the field robot built and datasets recorded for the evaluation of the plant classification system presented in Chapter 4 and plant position estimation pipeline developed in Chapter 5. Furthermore, it provides results and a thorough evaluation of both systems using the field data and the selected evaluation metrics.

The discussion of the presented novel pipelines and the combined system indicates that the objectives for this thesis are solved: The plant classification pipeline is able to process the vegetation segmented multispectral field images into full resolution plant classification images. For each pixel a predicted plant class is available. Using the plant position estimation system, plant positions for all plants (both crop and weed) are predicted in the image. The combined system is able to perform the task the farmer is most interested in: Single plant weed control. The system achieves the desired high weed treatment rates with only minimal loss of crop.

The methods cope well with the so far unsolved challenges in commercial organic carrot farms of generally small plants in the cm range, overlap between plants and a high variability of the plants appearance. The developed image acquisition, plant classification and stem estimation systems run on CPU in real time and can be applied on a moving robot to close the perceive-plan-act loop. This allows the realization of precision agriculture and especially the envisioned single plant weed control task.

7 Conclusion

Ensuring sustainable agriculture to feed a growing world population is a large ongoing worldwide challenge. Combining robotic capabilities with precision agriculture are a promising approach to make agricultural production more ecological, to increase productivity or improve cost. Especially in organic vegetable farming, weed control is essential to avoid yield losses of more than 50 % and the challenging task of intra-row weed control is still mostly performed manually.

This thesis develops a computer vision system for application in an agricultural robot which is able to detect and classify plants in the field: It applies machine learning to discriminate crop in challenging plant growing conditions with high levels of plant overlap and the presence of weeds of different sizes inside and outside the crop row. Additionally, a plant stem detection and position estimation method extracts the position of plants in the field with high precision.

Experiments with the custom built field robot Bonirob are carried out in commercial organic carrot farms. Several datasets with carrot crops in early growth stage are acquired with the field robot. In this stage, close-to-crop and intra-row weeds are present which are a challenge for the plant classification and weeding job. The dataset images are used to evaluate the proposed methods and to prove their performance and applicability to real world precision agricultural tasks. As no datasets in this challenging domain were publicly available, a dataset with plant class and vegetation segmentation annotations was published together with a paper.

The developed computer vision system comprises three major components: Image acquisition and vegetation segmentation, plant classification as well as plant detection and position estimation. As presented in the evaluation chapter these approaches fulfill the goals and research questions outlined in the introduction.

Multispectral Image Acquisition and Vegetation Segmentation A beam splitting multispectral camera setup is selected as best option to capture color and near-infrared images from a moving agricultural robot in the field. The camera setup delivers both time and spatially registered multispectral images with frame rates of 1 to 30 Hz. These images are well suited for the background removal step and the plant related image processing tasks because they capture the color and near-infrared channels which allows to leverage the red-edge property (low reflectance in red band and high reflectance in near-infrared band) of plants.

The developed vegetation segmentation method processes the multispectral images with a modified Normalized Difference Vegetation Index (NDVI) segmentation approach: Sev-

eral pre- and postprocessing steps ensure that different brightness levels of the RGB vs. NIR channel are corrected and that small segmentation artifacts and shadow effects are removed by filtering and blob size thresholding. The output of the vegetation segmentation step is a segmentation mask which covers all background (i.e. non vegetation) pixels. Using this mask, a masked NDVI image is created which is used in the following plant classification and position estimation pipelines.

Plant Classification The novel plant classification system processes the vegetation segmented NDVI images into full plant classification images. Each vegetation pixel in the output image is assigned to one of the different plant classes. The method is more generic than a crop/weed discrimination approach since it supports more than two plant classes. The system applies a two stage approach: During offline training human expert labeled ground truth is used to train the system in supervised mode. In the online application phase the system fully autonomously estimates the plant classification image for previously unseen images. The developed method supports real world field situations where crops and weeds are of difference sizes, grow close together and all types of plants overlap.

The plant classification pipeline does neither require nor apply segmentation into leaves or plants. It avoids this major source of error in previous systems and rather utilizes a divide and conquer approach: First, overlapping patches are extracted at sparsely distributed keypoints. Then, from each patch features are extracted (contour and shape features as well as NDVI value statistics) and a trained Random Forest classifier estimates plant classification scores for each patch. Second, the scores are smoothed using a Conditional Random Field (CRF) to yield more spatially smooth plant classification estimates. This step builds on the assumption that keypoints which are close together most likely belong to the same plant and therefore most likely have the same plant class. The CRF formulation is used to combine this smoothness term with a data term which is derived from the classifier scores. The output of the smoothing step is a smoothed categorical plant class label for each keypoint. Finally, the plant classification results which are until now only calculated per keypoints are interpolated back to full image resolution with the goal that for all vegetation pixels an estimated plant class is available.

This concludes the divide and conquer strategy and allows classification of challenging image regions with irregular shaped leaves as well as overlap where the plant segmentation based approaches are prone to fail. Additionally, the novel approach overcomes the loss of output precision of related work which only split the image into different regions (so-called cell-based methods) and just classify whether a cell belongs to the class plant or weed or background.

Visual inspection of the results shows the effectiveness of the pipeline which is able to produce plant classification images of high quality. The evaluation with ROC curves and classification metrics proves the performance quantitatively: Average accuracies of 91.4 % and 96.7 % are achieved by the plant classification system.

The plant classification image which is the output of the plant classification system can be directly used to treat weed patches with for example precision spraying of herbicide. Furthermore, the system's output can be used to address other agricultural tasks like plant

phenotyping: For example crop mapping and crop/weed coverage ratio calculations can be realized.

Plant Detection and Position Estimation The task of determining the position of a plant in the field is solved by a newly developed plant detection and position estimation pipeline. The vegetation segmented NDVI images are the only input data, the output of the system are detected plant positions.

A sliding window-based image patch extraction step yields image patches. The subsequent classification estimates whether such a patch displays a plant stem region or not. In the following, a filtering and non-maximum suppression step determines the discrete plant stem positions encoded as (u, v) pixel coordinates. The Random Forest plant stem classifier is trained in the offline phase using human expert labeled ground truth stem positions. In the online phase it can be applied directly without human intervention.

A thorough evaluation with visual inspection, confusion matrices and use of introduced position estimation metrics highlights the performance of the position estimation system: For the challenging position estimation threshold of 7.5 mm F1-scores of 78.6 % and 75.8 % are achieved; for a relaxed threshold of 20 mm the F1-score performance improves to 85.7 % and 83.8 % respectively.

This novel approach to plant position estimation has several key advantages over previous work: First, the system only requires the vegetation segmented images. Additional information leveraged by previous work such as knowledge of the row position/layout or all crop positions is not required. Second, it is able to produce plant position estimates for all plants, i.e. crops and weeds. The ability to also detect positions of individual weed plants allows single plant weed control and is a major improvement over related approaches with only deliver crop plant positions. Third, the plant detection and position estimation system does neither require a plant or leaf segmentation nor does it require extraction of plant structures like branches or leaf veins to derive the stem position. Such plant structure segmentation might work in simple cases with single plants but results in severe performance losses when applied to field images with many and overlapping plants.

The output of the plant position estimation step enables single plant precision agriculture and phenotyping measures such as plant counting.

Combined System and Farmer's Perspective The combination of both the presented plant classification and plant position estimation pipelines enables single plant precision agriculture which takes into account the plant class. For example on the one hand, the precise position of weed plants is needed in order to precisely target mechanical weeding tools. On the other hand, the position of crop plants is the required input for further phenotyping steps where for example plants are counted or single plant measurements like size or plant area of only crop plants are calculated.

An evaluation of the combined system provides results form a farmer's perspective. The estimated plant positions are combined with the plant classification information to derive treatment position for a weeding tool. The strict stem detection threshold of 7.5 mm is applied again and then the determined treatment positions are evaluated if they hit a

desired weed or accidentally remove a crop plant. On the two datasets A and B the system achieves the high weed treatment rate of 75.8 % with only minimal loss of 9.7 % crop. This proves the applicability of the combined system to solve the farmer's task of single plant mechanical weed control in organic crop farms where today typically still manual labor is necessary.

Finally, real world tests with the Bonirob field robot were conducted in commercial organic carrot farms within the publicly funded project RemoteFarming.1. The positions of the classified weed plants were estimated in real time and sent to a weeding tool. The weeding tool comprises a visual servoed delta robot with a tube stamp tool attached to the end-effector. The tube stamp was executed when the tool was positioned over weed plants to regulate the weed by pushing the plant several centimeters into the ground. Additionally, the presented approaches were transfered to the Bosch funded startup Deepfield Robotics. There, this research is used towards weed regulation in for example sugar beet fields.

All in all, the combination of the precise plant position and the plant type enables the realization of a multitude of precision agriculture tasks: This includes but is not limited to the presented mechanical weed control process. Hopefully, the use of this research combined with advanced robotics will enable fleets of intelligent agricultural robots to bring the world closer to the goal of achieving sustainable, ecological and human-friendly agricultural production.

A Additional Results

A.1 Results for the Crop Weed Field Image Dataset

In the following the plant classification results for the published Crop/Weed Field Image Dataset (CWFID) [70] are provided. This dataset was recorded on the same field as dataset A but only two classes crop and weed were labeled. For the presented evaluation the pipeline is parameterized with the same parameters as for dataset A. The concrete parameter values are given in Table 6.3. First, Figure A.1 displays the ROC curve achieved for the unsmoothed plant classification.

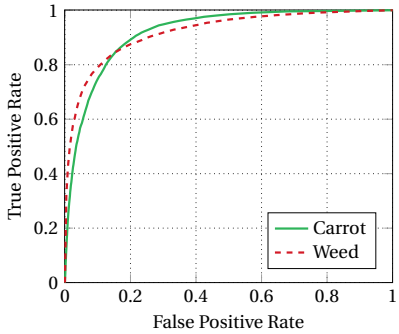


Figure A.1: Plant classification ROC curve for the CWFID dataset (before smoothing).

Second, Table A.1 indicates the positive effect of the smoothing process also for this dataset. All metrics improve substantially.

Table A.1: Improvement through smoothing on the CWFID dataset.

	Accuracy	Precision	Recall	F1-score
No Smoothing	86.9 %	81.7 %	78.7 %	80.1 %
After Smoothing	90.5 %	89.1 %	82.0 %	85.4 %
Improvement	+3.6 pp	+7.4 pp	+3.3 pp	+5.3 pp

Finally, Table A.2 gives the overall classification metrics after smoothing for the CWFID dataset.

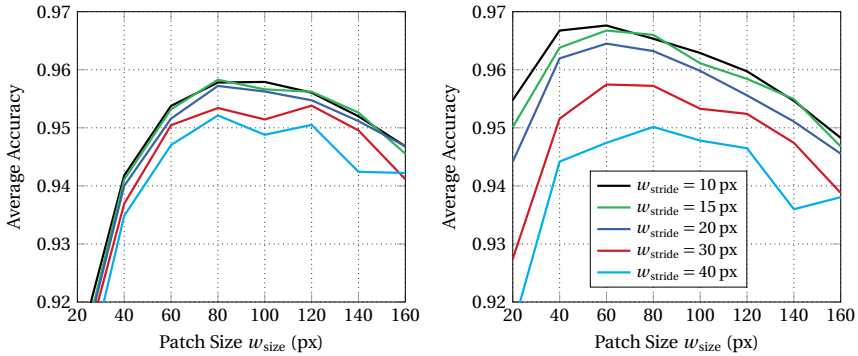
Table A.2: Final plant classification results (after smoothing) for the CWFID dataset.

CWFID Dataset	Accuracy	Precision	Recall	F1-score
Crop	90.5 %	86.9 %	66.8 %	75.5 %
Weed	90.5 %	91.3 %	97.2 %	94.1 %
Overall	90.5 %	89.1 %	82.0 %	85.4 %

It can be concluded that the presented pipeline improves compared to the published figures in [70] and that stable performance is achieved.

A.2 Plant Classification Parameter Selection for Dataset B

In the following the parameter selection for dataset B is plotted. The methodology is explained in Section 4.5 and the selected parameterization is given in Table 6.3.



(a) Before smoothing

(b) After smoothing

Figure A.2: Plant classification metrics for varying patch size w_{size} and patch stride w_{stride} before and after smoothing for dataset B.

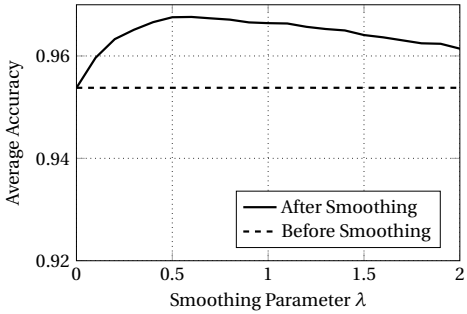


Figure A.3: Average accuracy when varying the smoothing parameter λ for plant classification on dataset B.

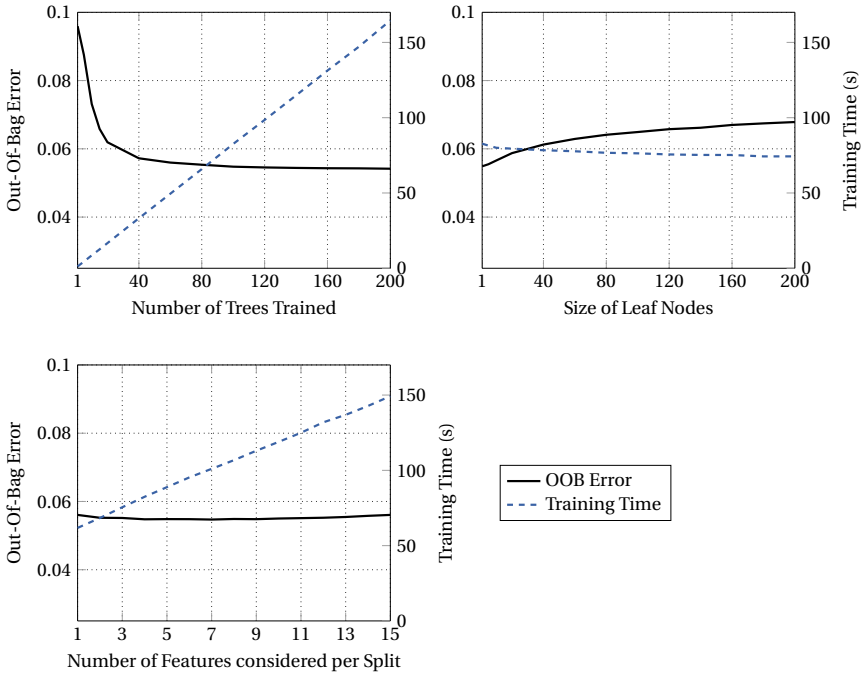


Figure A.4: Out-of-bag error of the Random Forest classifier (left axis) and training time (right axis) depending on the number of trees trained, leaf node size and number of features considered per split for plant classification on dataset B.

A.3 Plant Position Estimation Parameter Selection for Dataset B

This section presents the plant position estimation parameter selection for dataset B according to Section 5.5. The finally chosen parameterization is given in Table 6.7.

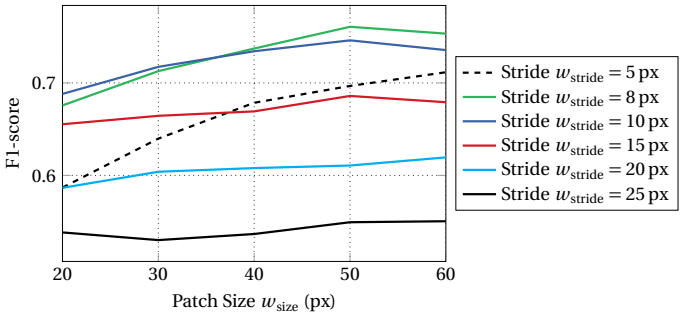


Figure A.5: Variation of patch size and patch stride for dataset B.

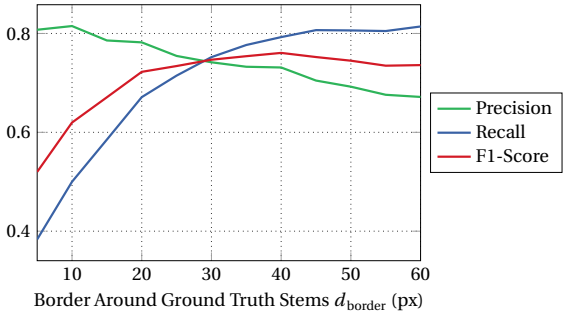


Figure A.6: Variation of the border around ground truth stems for dataset B.

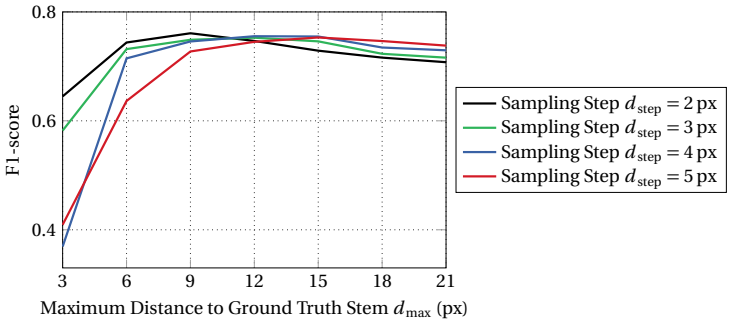


Figure A.7: Joint variation of the sampling step and the maximum distance where positive patches are extracted for dataset B.

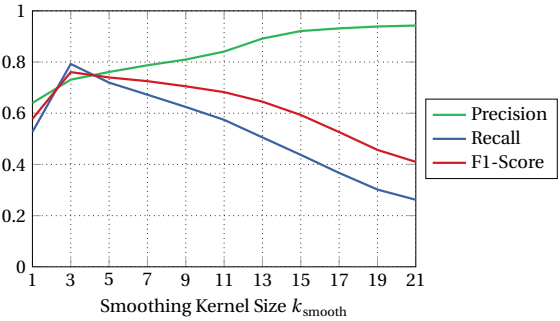


Figure A.8: Influence on plant stem detection metrics for different sizes of the smoothing kernel k_{smooth} for dataset B.

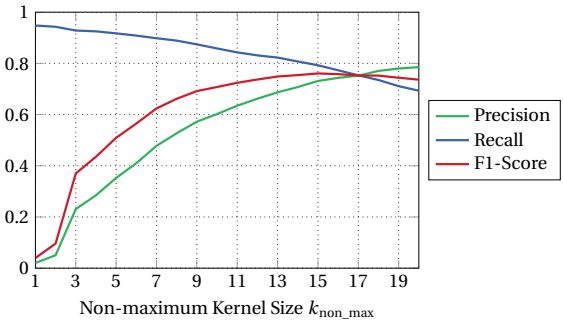


Figure A.9: Variation of the non-maximum suppression kernel size for dataset B.

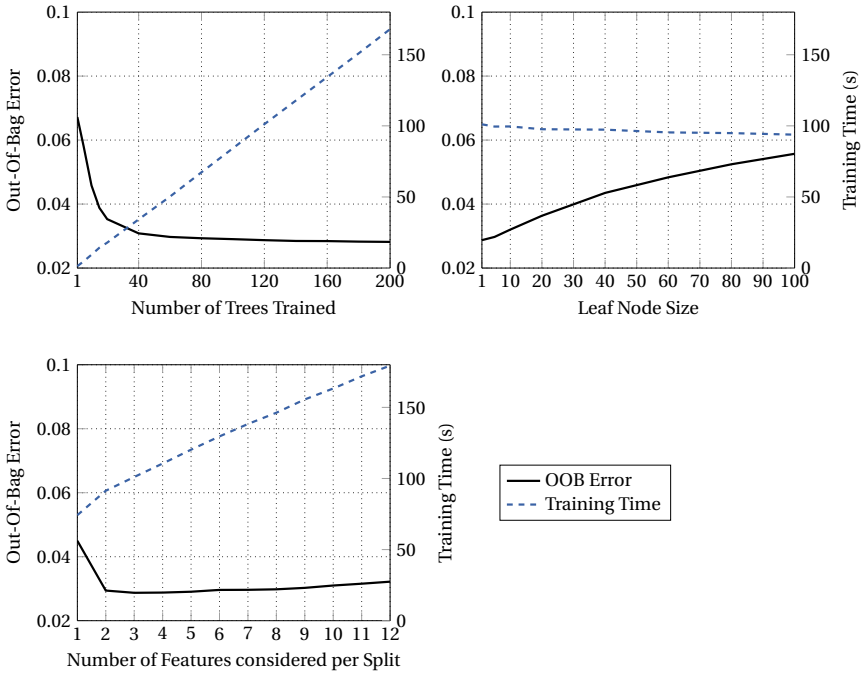


Figure A.10: Dataset B plant position estimation out-of-bag error of the Random Forest classifier (left axis) and training time (right axis) depending on the number of trees trained, leaf node size and number of features considered per split.

Bibliography

[1] N. Zhang, M. Wang, and N. Wang. Precision agriculture - a worldwide overview. *Computers and Electronics in Agriculture*, 36(2-3):113–132, 2002.

[2] S. Christensen, H. T. Søgaard, P. Kudsk, M. Nørremark, I. Lund, E. Nadimi, and R. Jørgensen. Site-specific weed control technologies. *Weed Research*, 49(3):233–241, 2009.

[3] O. Bawden, D. Ball, J. Kulk, T. Perez, and R. Russell. A lightweight, modular robotic vehicle for the sustainable intensification of agriculture. volume 02-04-December-2014, 2014.

[4] O. Bawden, J. Kulk, R. Russell, C. McCool, A. English, F. Dayoub, C. Lehnert, and T. Perez. Robot for weed species plant-specific management. *Journal of Field Robotics*, 34(6):1179–1199, 2017.

[5] J. Peña, J. Torres-Sánchez, A. 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(10), 2013.

[6] H. Laber and H. Stützel. Ertragswirksamkeit der restverunkrautung in gemüsekulturen nach nichtchemischen unkrautregulationsmaßnahmen. *Pflanzenbauwissenschaften*, 7(1):29–38, 2003.

[7] S. Fittje, M. Hänsel, F. Langsenkamp, A. Kielhorn, and D. Trautz. Praxiserhebungen zu Aufwand und Erfolg der Handjäte in Möhren unter ökologischer Bewirtschaftung. *13. Wissenschaftstagung Ökologischer Landbau*, 2015.

[8] S. B. Powles and Q. Yu. Evolution in action: plants resistant to herbicides. *Annual review of plant biology*, 61:317–347, 2010.

[9] R. Hillocks. Farming with fewer pesticides: EU pesticide review and resulting challenges for UK agriculture. *Crop Protection*, 31(1):85–93, 2012.

[10] R. Plant. Site-specific management: The application of information technology to crop production. *Computers and Electronics in Agriculture*, 30(1-3):9–29, 2001.

[11] C. Timmermann, R. Gerhards, and W. Kühbauch. The economic impact of site-specific weed control. *Precision Agriculture*, 4(3):249–260, 2003.

[12] W. Lee, V. Alchanatis, C. Yang, M. Hirafuji, D. Moshou, and C. Li. Sensing technologies for precision specialty crop production. *Computers and Electronics in Agriculture*, 74(1):2–33, 2010.

[13] C. Yang, J. Everitt, Q. Du, B. Luo, and J. Chanussot. Using high-resolution airborne and satellite imagery to assess crop growth and yield variability for precision agriculture. *Proceedings of the IEEE*, 101(3):582–592, 2013.

[14] M. Salas Fernandez, Y. Bao, L. Tang, and P. Schnable. A high-throughput, field-based phenotyping technology for tall biomass crops. *Plant Physiology*, 174(4):2008–2022, 2017.

[15] D. Slaughter, D. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1):63–78, 2008.

[16] J.-X. Du, D.-S. Huang, X.-F. Wang, and X. Gu. Computer-aided plant species identification (CAPSI) based on leaf shape matching technique. *Transactions of the Institute of Measurement and Control*, 28(3):275–285, 2006.

[17] J.-X. Du, X.-F. Wang, and G.-J. Zhang. Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 185(2):883–893, 2007.

[18] J. Liu, S. Zhang, and S. Deng. A method of plant classification based on wavelet transforms and support vector machines. volume 5754 of *Lecture Notes in Computer Science*, pages 253–260. 2009.

[19] T. Beghin, J. S. Cope, P. Remagnino, and S. Barman. Shape and texture based plant leaf classification. In *Advanced Concepts for Intelligent Vision Systems*, pages 345–353. Springer, 2010.

[20] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet. Combining leaf salient points and leaf contour descriptions for plant species recognition. In *Image Analysis and Recognition*, volume 7950 of *Lecture Notes in Computer Science*, pages 205–214. 2013.

[21] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012.

[22] Z.-Q. Zhao, L.-H. Ma, Y. ming Cheung, X. Wu, Y. Tang, and C. L. P. Chen. ApLeaf: An efficient android-based plant leaf identification system. *Neurocomputing*, 151, Part 3(0):1112–1119, 2015.

[23] J. C. Neto, G. E. Meyer, and D. D. Jones. Individual leaf extractions from young canopy images using gustafson–kessel clustering and a genetic algorithm. *Computers and Electronics in Agriculture*, 51(1):66–85, 2006.

[24] P. J. Komi, M. R. Jackson, and R. M. Parkin. Plant classification combining colour and spectral cameras for weed control purposes. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 2039–2042. IEEE, 2007.

[25] D. C. Slaughter, D. K. Giles, S. A. Fennimore, and R. F. Smith. Multispectral machine vision identification of lettuce and weed seedlings for automated weed control. *Weed Technology*, 22(2):378–384, 2008.

[26] A. Perez, F. Lopez, J. Benlloch, and S. Christensen. Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture*, 25(3):197–212, 2000.

[27] 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.

[28] J. Blasco, N. Aleixos, J. Roger, G. Rabatel, and E. Moltó. Robotic weed control using machine vision. *Biosystems Engineering*, 83(2):149–157, 2002.

[29] B. Åstrand and A.-J. Baerveldt. An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots*, 13(1):21–35, 2002.

[30] Y. Zheng, Q. Zhu, M. Huang, Y. Guo, and J. Qin. Maize and weed classification using color indices with support vector data description in outdoor fields. *Computers and Electronics in Agriculture*, 141:215–222, 2017.

[31] C. M. Onyango and J. Marchant. Segmentation of row crop plants from weeds using colour and morphology. *Computers and Electronics in Agriculture*, 39(3):141–155, 2003.

[32] C. Gée, J. Bossu, G. Jones, and F. Truchetet. Crop/weed discrimination in perspective agronomic images. *Computers and Electronics in Agriculture*, 60(1):49–59, 2008.

[33] 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*, pages 1–14, 2012.

[34] H. Suh, J. Hofstee, J. IJsselmuiden, and E. van Henten. Sugar beet and volunteer potato classification using bag-of-visual-words model, scale-invariant feature transform, or speeded up robust feature descriptors and crop row information. *Biosystems Engineering*, 166:210–226, 2018.

[35] M. Aitkenhead, I. Dalgetty, C. Mullins, A. McDonald, and N. Strachan. Weed and crop discrimination using image analysis and artificial intelligence methods. *Computers and Electronics in Agriculture*, 39(3):157–171, 2003.

[36] 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.

[37] A. Tellaeche, G. Pajares, X. P. Burgos-Artizzu, and A. Ribeiro. A computer vision approach for weeds identification through support vector machines. *Applied Soft Computing*, 11(1):908–915, 2011.

[38] H. Nejati, Z. Azimifar, and M. Zamani. Using fast fourier transform for weed detection in corn fields. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 1215–1219, Oct 2008.

[39] 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.

[40] K. Thorp and L. Tian. A review on remote sensing of weeds in agriculture. *Precision Agriculture*, 5(5):477–508, 2004.

[41] Y. Zhang, D. C. Slaughter, and E. S. Staab. Robust hyperspectral vision-based classification for multi-season weed mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 69(0):65–73, 2012.

[42] A. Wendel and J. Underwood. Self-supervised weed detection in vegetable crops using ground based hyperspectral imaging. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5128–5135, 2016.

[43] D. Stroppiana, P. Villa, G. Sona, G. Ronchetti, G. Candiani, M. Pepe, L. Busetto, M. Migliazzi, and M. Boschetti. Early season weed mapping in rice crops using

multi-spectral UAV data. *International Journal of Remote Sensing*, 39(15-16):5432–5452, 2018.

[44] M. Ehsani, S. Upadhyaya, and M. Mattson. Seed location mapping using RTK GPS. *Transactions of the American Society of Agricultural Engineers*, 47(3):909–914, 2004.

[45] H. Griepentrog, M. Nørremark, H. Nielsen, and B. Blackmore. Seed mapping of sugar beet. *Precision Agriculture*, 6(2):157–165, 2005.

[46] H. Sun, D. Slaughter, M. P. Ruiz, C. Gliever, S. Upadhyaya, and R. Smith. RTK GPS mapping of transplanted row crops. *Computers and Electronics in Agriculture*, 71(1): 32–37, 2010.

[47] B. Bennedsen, D. Peterson, and A. Tabb. Identifying defects in images of rotating apples. *Computers and Electronics in Agriculture*, 48(2):92–102, 2005.

[48] G. Polder, G. W. van der Heijden, J. van Doorn, and T. A. Baltissen. Automatic detection of tulip breaking virus (TBV) in tulip fields using machine vision. *Biosystems Engineering*, 117(0):35–42, 2014.

[49] V. Tejada, M. Stoelen, K. Kusnierek, N. Heiberg, and A. Korsaeth. Proof-of-concept robot platform for exploring automated harvesting of sugar snap peas. *Precision Agriculture*, 18(6):952–972, 2017.

[50] A. Ruckelshausen, L. Busemeyer, R. Klose, A. Linz, K. Moeller, M. Thiel, K. Alheit, F. Rahe, D. Trautz, and U. Weiss. Sensor and system technology for individual plant crop scouting. In *International Conference on Precision Agriculture (ICPA), 2010*, 2010.

[51] S. Dhondt, N. Wuyts, and D. Inzé. Cell to whole-plant phenotyping: the best is yet to come. *Trends in plant science*, 18(8):428–439, 2013.

[52] J. White, P. Andrade-Sanchez, M. Gore, K. Bronson, T. Coffelt, M. Conley, K. Feldmann, A. French, J. Heun, D. Hunsaker, M. Jenks, B. Kimball, R. Roth, R. Strand, K. Thorp, G. Wall, and G. Wang. Field-based phenomics for plant genetics research. *Field Crops Research*, 133:101–112, 2012.

[53] H. Søgaard and H. Olsen. Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 38(2):141–158, 2003.

[54] G. Jiang, Z. Wang, and H. Liu. Automatic detection of crop rows based on multi-ROIs. *Expert Systems with Applications*, 42(5):2429–2441, 2015.

[55] I. García-Santillán, J. Guerrero, M. Montalvo, and G. Pajares. Curved and straight crop row detection by accumulation of green pixels from images in maize fields. *Precision Agriculture*, 19(1):18–41, 2018.

[56] Y.-J. Huang and F.-F. Lee. An automatic machine vision-guided grasping system for phalaenopsis tissue culture plantlets. *Computers and Electronics in Agriculture*, 70 (1):42–51, 2010.

[57] H. S. Midtiby, T. M. Giselsson, and R. N. Jørgensen. Estimating the plant stem emerging points (PSEPs) of sugar beets at early growth stages. *Biosystems Engineering*, 111 (1):83–90, 2012.

[58] 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(4):755–765, 2012.

[59] C. Hunt, C. Jones, M. Hickey, J. Koolaard, J. West, and J.-H. Hatier. Estimation in the field of individual perennial ryegrass plant position and dry matter production using a custom-made high-throughput image analysis tool. *Crop Science*, 55(6): 2910–2917, 2015.

[60] M. Nørremark, H. Søgaard, H. Griepentrog, and H. Nielsen. Instrumentation and method for high accuracy geo-referencing of sugar beet plants. *Computers and Electronics in Agriculture*, 56(2):130–146, 2007.

[61] M. Nørremark, H. W. Griepentrog, J. Nielsen, and H. T. Søgaard. The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops. *Biosystems Engineering*, 101(4):396–410, 2008.

[62] 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.

[63] A. Nakarmi and L. Tang. Automatic inter-plant spacing sensing at early growth stages using a 3D vision sensor. *Computers and Electronics in Agriculture*, 82:23–31, 2012.

[64] D. Dey, L. Mummert, and R. Sukthankar. Classification of plant structures from uncalibrated image sequences. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 329–336, Jan 2012.

[65] G. Alenya, B. Dellen, and C. Torras. 3D modelling of leaves from color and tof data for robotized plant measuring. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3408–3414, May 2011.

[66] C. Bac, J. Hemming, and E. Van Henten. Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, 105: 111–120, 2014.

[67] K. Kusumam, T. Krajník, S. Pearson, T. Duckett, and G. Cielniak. 3d-vision based detection, localization, and sizing of broccoli heads in the field. *Journal of Field Robotics*, 34(8):1505–1518, 2017.

[68] S. Haug, P. Biber, A. Michaels, and J. Ostermann. Plant stem detection and position estimation using machine vision. In *Workshop Proceedings of IAS-13*, 13th Intl. Conf. on Intelligent Autonomous Systems, pages 483–490. 2014.

[69] S. Haug, A. Michaels, P. Biber, and J. Ostermann. Plant classification system for crop / weed discrimination without segmentation. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1142–1149. IEEE, 2014.

[70] S. Haug and J. Ostermann. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In *Computer Vision - ECCV 2014 Workshops*, volume 8928 of *Lecture Notes in Computer Science*, pages 105–116. 2015.

[71] S. Haug and J. Ostermann. Plant classification for field robots: A machine vision approach. In J. Zhou, X. Bai, and T. Caelli, editors, *Computer Vision and Pattern Recognition in Environmental Informatics*, pages 248–272. IGI Global, 2016.

[72] W. Bangert, A. Kielhorn, F. Rahe, A. Albert, P. Biber, S. Grzonka, S. Haug, A. Michaels, D. Mentrup, M. Hänsel, D. Kinski, K. Möller, A. Ruckelshausen, C. Scholz, F. Sellmann, et al. Field-robot-based agriculture: RemoteFarming.1 and BoniRob-Apps. *VDI Agricultural Engineering 2013*, pages 439–446, 2013.

[73] A. Michaels, R. Patil, S. Haug, and A. Albert. Monitoring grasping success in a robotic weed control application. *Journal of Agricultural Machinery Science*, 10(2):121–127, 2014.

[74] F. Sellmann, W. Bangert, S. Grzonka, M. Hänsel, S. Haug, A. Kielhorn, A. Michaels, K. Möller, F. Rahe, W. Strothmann, et al. RemoteFarming.1: Human-machine interaction for a field-robot-based weed control application in organic farming. In *Proceedings of 4th International Conference on Machine Control & Guidance (MCG)*, pages 36–42, 2014.

[75] A. Michaels, S. Haug, and A. Albert. Vision-based high-speed manipulation for robotic ultra-precise weed control. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5498–5505. IEEE, 2015.

[76] B. E. Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[77] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

[78] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.

[79] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11 (285–296):23–27, 1975.

[80] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sep 2004.

[81] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, Jun 2001.

[82] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[83] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[84] T. Tuytelaars, K. Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends in computer graphics and vision*, 3(3):177–280, 2008.

[85] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 1, pages 886–893, June 2005.

[86] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[87] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[88] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[89] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[90] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[91] V. Ila, L. Polok, M. Solony, and P. Svoboda. SLAM++-a highly efficient and temporally scalable incremental SLAM framework. *The International Journal of Robotics Research*, 36(2):210–230, 2017.

[92] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 14 of *IJCAI '95*, pages 1137–1145, 1995.

[93] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[94] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.

[95] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[96] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, Sep. 1999.

[97] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[98] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8): 861–874, 2006. ROC Analysis in Pattern Recognition.

[99] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[100] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997.

[101] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[102] R. P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, 27(11):47–50, Nov 1989.

[103] G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, Nov 2000.

[104] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[105] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone. Classification and regression trees. 1984.

[106] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[107] B. E. Boser, I. M. Guyon, and V. N. Vapnik. Training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[108] M. J. Kearns and Y. Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 98 of *ICML '98*, pages 269–277, 1998.

[109] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, Jan 2013.

[110] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[111] C. Sommer, C. Straehle, U. Kothe, and F. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 230–233, March 2011.

[112] M. Belgiu and L. Drăgu. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114: 24–31, 2016.

[113] J. Bohg, J. Romero, A. Herzog, and S. Schaal. Robot arm pose estimation through pixel-wise part classification. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3143–3150, 2014.

[114] D. Cutler, T. Edwards Jr., K. Beard, A. Cutler, K. Hess, J. Gibson, and J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.

[115] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, (2):139–157, 2000.

[116] T. Bylander. Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48(1-3):287–297, 2002.

[117] A. Cutler, D. Cutler, and J. Stevens. *Random forests*. 2012.

[118] B. Van Essen, C. Macaraeg, M. Gokhale, and R. Prenger. Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA? In *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, pages 232–239, April 2012.

[119] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7 (2–3):81–227, 2011.

[120] F. Baumann, A. Ehlers, K. Vogt, and B. Rosenhahn. Cascaded random forest for fast object detection. *Lecture Notes in Computer Science*, 7944 LNCS:131–142, 2013.

[121] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1393–1400, Sept 2009.

[122] A. Baldridge, S. Hook, C. Grove, and G. Rivera. The ASTER spectral library version 2.0. *Remote Sensing of Environment*, 113(4):711– 715, 2009.

[123] S. Nebiker, A. Annen, M. Scherrer, and D. Oesch. A light-weight multispectral sensor for micro UAV-opportunities for very high resolution airborne remote sensing. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 37, pages 1193–1199, 2008.

[124] C. Yang, J. Westbrook, C.-C. Suh, D. Martin, W. Hoffmann, Y. Lan, B. Fritz, and J. Goolsby. An airborne multispectral imaging system based on two consumer-grade cameras for agricultural remote sensing. *Remote Sensing*, 6(6):5257–5278, 2014.

[125] J. Downing, A. Murray, and A. Harvey. Low-cost multi-spectral imaging camera array. In *Computational Optical Sensing and Imaging, COSI 2012*, pages JW1A–6, 2012.

[126] S. Helling, E. Seidel, and W. Biehlig. Algorithms for spectral color stimulus reconstruction with a seven-channel multispectral camera. pages 254–258, 2004.

[127] J. Brauers, N. Schulte, and T. Aach. Multispectral filter-wheel cameras: Geometric distortion model and compensation algorithms. *IEEE Transactions on Image Processing*, 17(12):2368–2380, Dec 2008.

[128] R. Berns, L. Taplin, M. Nezamabadi, M. Mohammadi, and Y. Zhao. Spectral imaging using a commercial color-filter array digital camera. *ICOM-CC 14th Triennial Meeting*, pages 743–750, 2005.

[129] J. Antila, R. Mannila, U. Kantojärvi, C. Holmlund, A. Rissanen, I. Näkki, J. Ollila, and H. Saari. Spectral imaging device based on a tuneable MEMS Fabry-Perot interferometer. *Proceedings of SPIE - The International Society for Optical Engineering*, 8374, 2012.

[130] M. Abuleil and I. Abdulhalim. Narrowband multispectral liquid crystal tunable filter. *Optics Letters*, 41(9):1957–1960, 2016.

[131] J.-I. Park, M.-H. Lee, M. D. Grossberg, and S. K. Nayar. Multispectral imaging using multiplexed illumination. In *Computer Vision (ICCV), 2007 IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[132] M. Parmar, S. Lansel, and J. Farrell. An LED-based lighting system for acquiring multispectral scenes. In *IS&T/SPIE Electronic Imaging*, pages 82990–82990. International Society for Optics and Photonics, 2012.

[133] M. Kise, B. Park, G. Heitschmidt, K. Lawrence, and W. Windham. Multispectral imaging system with interchangeable filter design. *Computers and Electronics in Agriculture*, 72(2):61–68, 2010.

[134] X. Cao, X. Tong, Q. Dai, and S. Lin. High resolution multispectral video capture with a hybrid camera system. pages 297–304, June 2011.

[135] Z. Chen, X. Wang, and R. Liang. RGB-NIR multispectral camera. *Optics Express*, 22 (5):4985–4994, 2014.

[136] JAI. Datasheet: JAI AD-130GE 2 CCD Multispectral Camera. Document version: 03/2012, retrieved: 01/2020.

[137] D. Woebbecke, G. Meyer, K. Von Bargen, and D. Mortensen. Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASAE*, 38(1):259–269, 1995.

[138] G. Meyer and J. Neto. Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture*, 63(2):282–293, 2008.

[139] X. Tang, M. Liu, H. Zhao, and W. Tao. Leaf extraction from complicated background. In *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pages 1–5, Oct 2009.

[140] X. Bai, Z. Cao, Y. Wang, Z. Yu, Z. Hu, X. Zhang, and C. Li. Vegetation segmentation robust to illumination variations based on clustering and morphology modelling. *Biosystems Engineering*, 125:80–97, 2014.

[141] H. Suh, J. Hofstee, and E. van Henten. Shadow-resistant segmentation based on illumination invariant image transformation. In *Proceedings International Conference of Agricultural Engineering*, 2014.

[142] L. Zheng, J. Zhang, and Q. Wang. Mean-shift-based color segmentation of images containing green vegetation. *Computers and Electronics in Agriculture*, 65(1):93–98, 2009.

[143] L. Zheng, D. Shi, and J. Zhang. Segmentation of green vegetation of crop canopy images based on mean shift and fisher linear discriminant. *Pattern Recognition Letters*, 31(9):920–925, 2010.

[144] W. Guo, U. 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.

[145] K. Keller, N. Kirchgessner, R. Khanna, R. Siegwart, A. Walter, and H. Aasen. Soybean leaf coverage estimation with machine learning and thresholding algorithms for field phenotyping. *Proceedings of BMVC 2018*, page 0032, 2018.

[146] Y. Campos, E. Rodner, J. Denzler, H. Sossa, and G. Pajares. Vegetation segmentation in cornfield images using bag of words. *Lecture Notes in Computer Science*, 10016: 193–204, 2016.

[147] J. Romeo, G. Pajares, M. Montalvo, J. Guerrero, M. Guijarro, and J. De La Cruz. A new expert system for greenness identification in agricultural images. *Expert Systems with Applications*, 40(6):2275–2286, 2013.

[148] Y. Suzuki, H. Okamoto, and T. Kataoka. Image segmentation between crop and weed using hyperspectral imaging for weed detection in soybean field. *Environment Control in Biology*, 46(3):163–173, 2008.

[149] J. Marchant, H. Andersen, and C. Onyango. Evaluation of an imaging sensor for detecting vegetation using different waveband combinations. *Computers and Electronics in Agriculture*, 32(2):101–117, 2001.

[150] D.-V. Nguyen, L. Kuhnert, and K. Kuhnert. Structure overview of vegetation detection. a novel approach for efficient vegetation detection using an active lighting system. *Robotics and Autonomous Systems*, 60(4):498–508, 2012.

[151] I. Scotford and P. Miller. Applications of spectral reflectance techniques in northern european cereal production: a review. *Biosystems Engineering*, 90(3):235–250, 2005.

[152] C. L. Wiegand, A. J. Richardson, D. E. Escobar, and A. H. Gerbermann. Vegetation indices in crop assessments. *Remote Sensing of Environment*, 35(2):105–119, 1991.

[153] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[154] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (4):509–522, 2002.

[155] H. Goëau, P. Bonnet, A. Joly, I. Yahiaoui, D. Barthelemy, N. Boujemaa, and J.-F. Molino. The ImageCLEF 2012 plant identification task. *CEUR Workshop Proceedings*, 1178, 2012.

[156] K. Choi, S. Han, S. Han, K.-H. Park, K.-S. Kim, and S. Kim. Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields. *Computers and Electronics in Agriculture*, 113:266–274, 2015.

[157] S. Janitza, G. Tutz, and A.-L. Boulesteix. Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics and Data Analysis*, 96:57–73, 2016.

[158] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, 2001.

[159] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[160] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11): 1222–1239, Nov 2001.

[161] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, Sept 2004.

[162] B. Scheuermann and B. Rosenhahn. Slimcuts: Graphcuts for high resolution images using graph reduction. *Lecture Notes in Computer Science*, 6819 LNCS:219–232, 2011.

[163] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77 (1–3):157–173, 2008.

[164] K. Yamamoto, W. Guo, Y. Yoshioka, and S. Ninomiya. On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors (Switzerland)*, 14(7):12191–12206, 2014.

[165] R. Raja, D. Slaughter, S. Fennimore, T. Nguyen, V. Vuong, N. Sinha, L. Tourte, R. Smith, and M. Siemens. Crop signalling: A novel crop recognition technique for robotic weed control. *Biosystems Engineering*, 187:278–291, 2019.

[166] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[167] A. Ruckelshausen, P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, F. Rahe, R. Resch, M. Thiel, D. Trautz, et al. Bonirob – an autonomous field robot platform for individual plant phenotyping. *Precision Agriculture*, 9:841, 2009.

[168] U. Weiss and P. Biber. Semantic place classification and mapping for autonomous agricultural robots. In *Proceeding of IROS Workshop on Semantic Mapping and Autonomous Knowledge Acquisition*, 2010.

[169] P. Biber, U. Weiss, M. Dorna, and A. Albert. Navigation system of the autonomous agricultural robot bonirob. In *Workshop on Agricultural Robotics: Enabling Safe,*

Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal, 2012.

[170] C. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, O. Bawden, R. Russell, and T. Perez. Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters*, 3(2):1184–1190, 2018.

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

[172] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[173] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[174] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, jan 2015.

[175] M. Firman. RGBD datasets: Past, present and future. In *CVPR Workshop on Large Scale 3D Data: Acquisition, Modelling and Analysis*, 2016.

[176] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[177] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[178] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LIDAR sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, pages 9297–9307, 2019.

[179] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[180] O. Söderkvist. Computer vision classification of leaves from swedish trees. Master's thesis, Linköping University, Sweden, 2001.

[181] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007.

[182] G. Agarwal, P. Belhumeur, S. Feiner, D. Jacobs, W. J. Kress, R. Ramamoorthi, N. A. Bourg, N. Dixit, H. Ling, D. Mahajan, et al. First steps toward an electronic field guide for plants. *Taxon*, 55(3):597–610, 2006.

[183] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition letters*, 81:80–89, 2016.

Lebenslauf

Sebastian Alexander Haug
geboren am 24. Oktober 1986
in Stuttgart

Beruf

seit 04/2019	Gruppenleiter bei der Robert Bosch GmbH Corporate Research, Robot Navigation and Perception
07/2015 – 03/2019	Forschungsingenieur bei der Robert Bosch GmbH Corporate Research, Robotic Systems and Power Tools
07/2012 – 08/2020	Doktorand bei Leibniz Universität Hannover Institut für Informationsverarbeitung (TNT) Prof. Dr.-Ing. Jörn Ostermann bis 06/2015 im Doktorandenprogramm bei Robert Bosch GmbH, Corporate Research Future Systems Consumer Goods
07/2010 – 01/2011	Praktikum bei Robert Bosch LLC Palo Alto, California, USA Autonomous Systems and Robotics Group

Ausbildung

10/2006 – 04/2012	Studium Elektrotechnik und Informationstechnik an der Universität Stuttgart Abschluss: Dipl.-Ing. mit Auszeichnung
03/2011 – 06/2011	Auslandssemenster an der Uppsala Universitet Uppsala, Schweden
09/1997 – 07/2006	Gymnasium in Stuttgart Abschluss: Abitur

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-387010-3