Deep learning-based visual recognition of rumex for robotic precision farming

Kounalakis, Tsampikos; Triantafyllidis, Georgios A.; Nalpantidis, Lazaros

Published in:
Computers and Electronics in Agriculture

Link to article, DOI:
10.1016/j.compag.2019.104973

Publication date:
2019

Document Version
Early version, also known as pre-print

Link back to DTU Orbit

Citation (APA):

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

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

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Deep learning–based visual recognition of rumex for robotic precision farming

Article in Computers and Electronics in Agriculture · October 2019
DOI: 10.1016/j.compag.2019.104973

CITATIONS
6

READS
180

3 authors, including:

Tsampikos Kounalakis
Danish Technological Institute
17 PUBLICATIONS 54 CITATIONS

See profile

Lazaros Nalpantidis
Technical University of Denmark
74 PUBLICATIONS 1,340 CITATIONS

See profile

Some of the authors of this publication are also working on these related projects:

Autonomous marine surface vessels (ShippingLab Autonomy) View project

STAMINA View project
Deep Learning-based Visual Recognition of Rumex for Robotic Precision Farming

Tsampikos Kounalakis\textsuperscript{a}, Georgios A. Triantafyllidis\textsuperscript{b}, Lazaros Nalpantidis\textsuperscript{c}

\textsuperscript{a}Service Robotics, Department of Robot Technology, Danish Technological Institute, Odense, Denmark
\textsuperscript{b}Department of Architecture, Design and Media Technology, Aalborg University, Copenhagen, Denmark
\textsuperscript{c}Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark

Abstract

In this paper we address the problem of recognising the Broad-leaved dock (Rumex obtusifolius L.) in grasslands from high-resolution 2D images. We discuss and present the determining factors for developing and implementing weed visual recognition algorithms using deep learning. This analysis, leads to the formulation of the proposed algorithm. Our implementation exploits Transfer Learning techniques for deep learning-based feature extraction, in combination with a classifier for weed recognition. A prototype robotic platform has been used to make available an image dataset from a dairy farm containing broad-leaved docks. The evaluation of the proposed algorithm on this dataset shows that it outperforms competing weed/plant recognition methods in recognition accuracy, while producing low false-positive rates under real-world operation conditions.

Key words: Weed recognition, Deep learning visual recognition, Agricultural robotics, Precision farming

Email addresses: tsko@teknologisk.dk (Tsampikos Kounalakis), gt@create.aau.dk (Georgios A. Triantafyllidis), lanalpa@elektro.dtu.dk (Lazaros Nalpantidis)
1. INTRODUCTION

Farming constitutes an important socio-economic activity all around the world. With an ever-growing market, agricultural research and industry seeks ways of making future agricultural production sustainable. Therefore, precision farming—often employing agricultural robotic systems—focuses on observing and treating individual part of farms or crops according to their individual needs.

One of the most promising cases of robotized precision farming is weed control. Therefore, a lot of attention has been given by the scientific community and several weed control robotic systems have been presented using Unmanned Aerial Vehicles (UAV) \cite{1,2} and Unmanned Ground Vehicles (UGV) \cite{3,4,5}. Such systems rely on accurate and reliable recognition of weeds from visual or other types of sensory data \cite{1,3,6,4,7,8,9,10,11,12,13}. Such weed control systems are able to detect plants that are not part of the preferred flora and apply treatment to individual plants, e.g. either by precision spraying or by physical removal of the plant. As a result, minimum or no collateral damage is done to the agricultural production and the environment. Furthermore, a more efficient farming paradigm is achieved, where the minimum amount of resources, such as water, energy or pesticides, are used.

Our envisioned solution \cite{14} is an autonomous precision farming robot that facilitates organic farming in dairy farms. The robot integrates visual weed recognition from high-resolution 2D images captured in the uncontrolled environment of grasslands. While the majority of relevant literature focuses on crops, the much more challenging setting of grasslands has started being

![Figure 1: Left: Open-field image acquisition prototype robotic platform. Top right: View from the robot camera containing a Rumex plant. Bottom right: Recognition results of our proposed algorithm.](image)

Our envisioned solution \cite{14} is an autonomous precision farming robot that facilitates organic farming in dairy farms. The robot integrates visual weed recognition from high-resolution 2D images captured in the uncontrolled environment of grasslands. While the majority of relevant literature focuses on crops, the much more challenging setting of grasslands has started being
explored as well [7, 8, 9, 10, 11, 12, 13]. The weed to be recognized by our proposed system is the Broad-leaved dock (Rumex obtusifolius L.), a weed with wide geographic distribution that constitutes a major problem for dairy farms in many parts of the world. Its low nutritional value, as well as, its high contents of oxalic acid and oxalates can lead to animal health issues, if consumed by the animals in high quantities, while hindering the quality of products. Its spread can lead to a loss of usable pasture areas, and its usual herbicide treatment can hinder the diversity of multi-species pastures.

Fig. 1 illustrates the open-field image acquisition prototype robotic platform used for this work, as well as the output of the proposed weed recognition system.

The goal of this work is to develop the visual recognition system of an autonomous robotic solution that enables the detection of Rumex weed plants in grasslands. In this manner, this work contributes ultimately towards reduction of pesticide use and manual labor for weed control. Our hypothesis is that the combination of pre-trained deep neural networks for feature extraction with typical machine learning classifiers can provide detection rates only achievable by deep learning, but without the need for extensive, annotated datasets—as it is only the classifier that needs to be trained on relevant data.

The contribution of this paper is twofold. First, we carefully analyze the theoretical and practical aspects of designing and implementing deep learning-based weed recognition systems for real world weed recognition systems. Second, based on of the aforementioned analysis, we propose a deep learning-based visual recognition system using high-resolution 2D images and exploiting the concept of Transfer Learning, that achieves better than state-of-the-art performance when evaluated on recognizing Rumex plants in grasslands under real conditions, such as strongly imbalanced training datasets, challenging lighting and other environmental conditions.

Differently from the vast majority of related literature in weed recognition, our proposed system addresses the task of weed recognition in grasslands, with no clear color differentiation between the weeds and the surrounding flora or background, constituting our problem very challenging. Therefore, several design considerations contributed in the final pipeline of the proposed algorithm. Each considered frame acquired from a top-view camera mounted on the robot [13], is partitioned in several blocks, corresponding to spatial regions, using the proposed experimental framework in [10]. In order to perform the recognition process and compensate for limitation of deep learning techniques to process high-resolution images, we exploit
Transfer Learning, where the content of each image is represented with features computed from a pre-trained Convolutional Neural Network (CNN) [15, 16, 17, 18, 19]. The use of CNN as feature extractors [20, 16, 21, 22] can output robust and discriminative features even for images captured under challenging illumination and environmental conditions.

The objective of this work was to achieve the highest possible recognition rate, while keeping False-Positive Rate (FPR) and False-Negative Rate (FNR) as low as possible. A robotic system for weeding should primarily produce low false-positive results, i.e. avoid to falsely indicate the existence of a weed, which does not exist in reality. The reason is that in such cases the system is wasting time and resources for treating non-existent targets, adversely affecting the robot’s autonomy. On the other hand, the existence of false-negative results, i.e, the number of existing weeds misclassified as soil or grass, is not considered as critical and can be compensated by multiple robot passes. The experimental assessment is performed on the publicly available Rumex Ancenis dataset acquired in an open-field in France [13].

The rest of the paper is organized as follows, in Section 2 we review related work from recent weed control robotic systems and deep learning-based plant recognition methods. Section 3 presents the theoretical and practical considerations behind the development of our weed recognition algorithm presented in Section 4. Moreover, in Section 5 we describe the performance of our proposed weed recognition system, as shown by a detailed experimental assessment. Finally, conclusions are drawn in Section 6.

2. RELATED WORK

Most reported robotic systems use weed recognition algorithms based on man-defined features able to represent image content information [1, 3, 6, 4, 7, 10, 12, 11]. Specifically for the weed recognition of the Broad-leaved dock, the work in [7] proposed the use of texture information for real-time detection of the Broad-leaved dock in grasslands. Texture information is retrieved by means of a 2D-Fourier analysis on spatial regions where their total spectral power indicates the detection of weed or grass. Experiments performed under laboratory conditions and field testing show effective detection. However, spectral-based textural information lacks robustness under variant natural

1Rumex Ancenis dataset: https://dtudk-my.sharepoint.com/:u:/g/personal/lanalpa_win_dtu_dk/EUrah9-sm-pKn_6xvU8avgg5Xvdh-bnImuGZoK6wRzPTrQ?e=OMKbLv
illumination conditions, e.g., shades, which can affect the thresholds set for detection.

The weed recognition framework in [10] was based on hand crafted feature-based object and image categorization systems. That framework allowed combinations between feature extraction and detection methods, as well as methods that contribute to the improvement of their discriminatory capabilities. The resulting image representations can be also combined with various classifiers. That framework was used for developing a weed recognition system for the Broad-leaved dock [10] based on SIFT features. The same framework was also used in [12] for the development of a system capable of recognizing thistles in sugar beets plantations [6]. The resulting system, also based on the SURF descriptor, was shown to outperform competing methodologies on the examined dataset. Moreover, the same system architecture was again applied for recognizing Broad-leaved dock yielding improved recognition results compared to the system in [10].

However—as it will also be experimentally shown in Section 5.6—the main limiting factor of weed recognition systems, such as [10], [11], [12], is the limited ability of their considered hand-crafted features to effectively represent image content. A strength of Deep learning-based algorithms is that image features are not man-designed, but rather learned. Commonly, deep learning algorithms typically achieve content representations that outperform traditional features in many visual recognition tasks such as image recognition [20], [16], [22] and segmentation [23]. As a result, researchers on the topic of weed or plant detection developed systems incorporating deep learning methodologies.

In [24] the authors propose a deep learning-based robotic weed detection system for sugar beet crops. They use a CNN for plant detection using as input near-infrared image frames from a 4-channel camera mounted on a robot. The detected plants are then recognized using a two-layer CNN. The sequence of CNNs combined with a blob-wise based voting, yields real-time detection with comparable results to the hand crafted feature-based methodology proposed in [4]. The method in [24] is restricted by the fact that its specialized detection is not always applicable in every application, e.g., detection in grasslands. Moreover, due to the lack of available training data the proposed CNN does not yield performance advantages over the competing hand crafted feature-based method.

Another application of CNNs is the fine-grained plant classification, which describes the classification problem between species of plants, i.e., intra-class
variations.

In [21], authors perform the method of Transfer learning [25] to participate in the plant identification task of the LifeCLEF challenge [26]—a challenge on recognizing images that depict living species. That proposed methodology is using CNN-computed feature representations which are combined with an extreme randomized tree classifier [27] yielding top-recognition results. This is achieved by using the Overfeat CNN architecture presented in [28] which is also trained for the image categorization task of the ImageNet dataset [29].

The same classification problem was also investigated in [30], which applies another form of transfer learning. That system uses the CNN proposed in [15], which was originally developed for performing image categorization on the ImageNet dataset [29]. This is done by removing the decision layer of the aforementioned CNN, i.e, the softmax layer, and re-initializing it with random values and the appropriate number of classes for the examined dataset. After a training session, the softmax layer provides top-performing results for the LifeCLEF dataset. The works of [21] and [30] are both based on the classification of images with little or no background clutter.

Recently, [31] implemented CNNs for semantic segmentation. In their paper, the authors implement a fully connected CNN. However, even in this approach acquired images are scaled (384×512 pixels) in order to be used with current deep learning methods. Thus, high-resolution images and their increased image information are not fully exploited while most of visual information is discarded because of the rescaling process.

In this work we investigate how to maximize the performance yield of deep learning-based methodologies, and make them applicable to our specific problem of recognizing Rumex plants. As an outcome of reviewing related literature, we have identified that the state-of-the-art methods in [10, 12, 21] and [30] are the most relevant ones for the problem we consider in this work, and thus use them for comparing the performance of our proposed system.

3. Design considerations for deep learning-based systems

The advancements in deep learning methods have resulted in remarkable results on the tasks of visual detection and recognition. However, the plethora of available choices, does not necessarily make the design of such a system easier. In this section, we discuss the possible design decisions and
Table 1: General considerations regarding the design of deep learning-based systems using high resolution images.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Sub-topic</th>
<th>Design Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image resolution</td>
<td></td>
<td>For the task of recognition and detection high resolution images are advised. Increased recognition performance comes at the expense of increased computational requirements.</td>
</tr>
<tr>
<td>Region description</td>
<td>Sliding window</td>
<td>Needs to be defined based on system requirements and recognition performance. Can be highly parallelized. Can segment high-resolution images to computationally tractable smaller regions.</td>
</tr>
<tr>
<td></td>
<td>Keypoint extraction</td>
<td>Creates sparse representation with increased computational efficiency. However, currently there is not strong research outcomes towards the combination of keypoint extraction and deep learning.</td>
</tr>
<tr>
<td></td>
<td>Region proposals</td>
<td>A very efficient way for detection. Used from various object detection methods. Can not be applied for highly detailed, high resolution images.</td>
</tr>
<tr>
<td>Training data generation</td>
<td>Increased autonomy</td>
<td>Image training ratio must be set in a way that false positives should be avoided.</td>
</tr>
<tr>
<td></td>
<td>Recognition accuracy</td>
<td>the ratio between classes must be set even.</td>
</tr>
</tbody>
</table>
analyze their characteristics for real-world visual recognition systems. The main points of this analysis are summarized in Table 1.

3.1. Image Resolution

The choice of image resolution, and therefore the camera sensor and lens can affect the real-time performance of the system, especially in deep learning applications. Most relevant deep learning algorithms [15, 16, 17, 18, 19] are only enabling the use of images with resolution of $224 \times 224$ pixels. This hinders the visual ability of systems, especially in real world conditions. Thus, the system developer must setup or use a sensor, taking into regard the balance between visual information and computational efficiency. In general, high resolution is advised, because it can promote improved image feature extraction. The increased computational load can then be tackled with the use of region proposal techniques.

3.2. Description of Regions-of-interest

The basic approach for region description is the sliding window [13]. This method can be very efficient especially when parallelized, but the performance depends on the number of sliding windows. This technique is highly customizable in terms of window resolution and window overlap, i.e. when a region is overlapping with the one in a previous position. The sliding window approach is the best suited for deep learning-based grassland robotic applications for a number of reasons. First, sliding windows are preferred because it is almost impossible to segment between foreground and background, i.e, segmentation of weed plants and grass. Then, in contrast to methods that use key-points or region proposal, which will add additional computation steps, sliding window can contribute in achieving real-time performance of weed control implementations in grasslands. The work in [28] shows that
sliding window can be integrated in the architecture of a CNN, were it would be further accelerated by GPU computing.

Keypoint or region proposal methods can enable accurate detection but they come at an increased computation cost. Keypoint methods extract features from parts of the image that are robust to rotation, illumination and blurriness. The application of this method in a grassland-based application can result in large amount of features and thus computation overhead due to the high number of proposed feature keypoints. That is not the case, e.g, in weed recognition in crops, where an initial segmentation method, e.g, using NDVI [30], forces keypoints to be detected on important plant parts. However, the implementation of deep learning keypoint-based methods is quite limited [32, 33].

Deep region proposals [34, 35], are the most common method for deep learning image-based detection. These methods propose multiple image regions which are considered as candidates for containing important information. Based on the expected detection accuracy, i.e, position of predicted regions, the user can define a threshold of what is a satisfactory overlap between predicted and ground truth regions of interest, i.e, Intersection over Union (IoU). These detectors have been shown to work well in specific datasets. However, their main disadvantage lies on the amount of available annotated data for training, robustness under various illumination conditions and that their current architectures do not allow their implementation in high resolution images.

3.3. Considerations for Deep Learning Training Data

The availability of data, due to the natural distribution of plants in grasslands and pastures, constitutes an additional problem when considering deep learning methodologies. Any dataset captured in grasslands and pastures is almost certainly expected to have a very small portion of images depicting a specific weed, with respect to the total amount of images in the dataset. However, this fact constitutes a big problem especially for data intensive learning tasks, such as deep learning. The reason is that the balance ratio between classes in datasets plays a significant role in the overall recognition or detection performance.

In crop-based applications, the distribution of plants is somewhat controllable, i.e, series of plants in predefined positions. Moreover, it is expected to have a satisfactory number of examples both for the crop and weed plant. On the other hand, in the case of grassland applications this distribution is not
defined and is close to random. For example, in the examined dataset due to the distribution of the weed plants in the field, for every image depicting the Broad leaved dock there are 80 images only containing background, i.e. grass, soil, etc. This introduces problems to the annotation process and it introduces bias to the classification model.

Therefore, a trained model can be overwhelmed by the dominant class, i.e., background, and not be able to recognize any other class, i.e., weed plant. Choosing a proper ratio between the two classes, as experimentally shown in Section 5.4, has a serious effect on a robotic system’s autonomy. If autonomy is of high importance then the users should consider to influence their deep learning model in order to prevent the occurrence of false positive predictions. In case that accuracy is of utmost importance then a 1:1 ratio between classes should be imposed during the classifier training phase.

4. PROPOSED VISUAL WEED RECOGNITION SYSTEM

The process of applying already trained CNNs and their resulting representations to other cases falls within the topic of Transfer learning. Using transfer learning with well-known deep learning architectures has experimentally shown state-of-the-art results in several topics, such as grasping, pose estimation, fine-grained plant categorization and object/image categorization. However, the adaptation of existing CNN models for use in such cases is not always straightforward, as additional case-specific adaptations are often required.

In this paper, we propose an effective application of transfer learning to a high resolution robotic weed recognition system, used to recognize the existence of Broad-leaved docks in grasslands. The overview of our proposed system is given in Fig. 2 and explained below. After the image acquisition process—that will be discussed in further details in Sec. 5.1—images are segmented using a 3x3 grid during the dataset annotation, as described in [10]. Thereafter, a CNN is used for calculating image representations and these features are fed into a classifier.

4.1. CNN-based feature extraction

Fine-tuning a pre-trained CNN model for a new classification problem requires the re-initialization of several layers in the network, i.e., the latter of the fully connected layers and the softmax layer which are responsible for the classification task. This process can be trivial in well-defined and structured
datasets, but it comes with caveats when applied in realistic robotic visual recognition tasks.

During fine-tuning, the network is going through a new training phase where a small learning rate is set for the convolutional layers, thus slightly affected by the new data, and a large learning rate for the re-initialized layers. Overfitting the newly adapted CNN can be easily achieved due to the increased learning rate of the reinitialized layers in combination with the small amount of training images, which is usually the case in datasets of field robotics.

Exploiting the advances of deep CNNs for object/image categorization [15, 16, 17, 18, 19], our proposed system is using the technique known as CNN-based feature extraction [20, 16, 21, 22]. This technique uses activations of intermediate CNN layers that can be combined with any classifier. For more details regarding the visualization of weights and patterns of CNN, we refer the interested reader to [37]. The pre-trained intermediate CNN layers, extract features that provide rich and robust image information to be used as input vectors for classification. This will allow us to describe images with CNN computed features without having to fine-tune a pre-trained CNN [25] or train a new model.

4.2. Classification

The resulting CNN-computed feature vectors can then be used in conjunction with any classifier. The classifier is able to learn and make decisions, thus substituting the operation of a softmax layer.

Since the examined classification problem is between two classes we investigate the use of a logistic regression algorithm, which can effectively classify CNN-extracted features without the added complexity or aforementioned disadvantages of CNN fine-tuning. Support Vector Machines (SVMs) can provide top-performing recognition results for a variety of applications [20, 16, 21, 22, 10, 11, 12]. We also consider these classifiers for our proposed system because they are able to cope with high-dimensional vectors, such as those derived from a CNN. Additionally, the convergence and performance yields of the examined classifiers are less dependant to the amount of available data, in contrast to deep learning methodologies, making them an excellent fit for the examined classification problem.

Specifically, in this work we only consider linear classifiers [38] for additional advantage of lesser computational complexity in comparison to non-linear classifiers while still resulting in state-of-the-art results for weed recog-
Figure 3: Example frames from the challenging night operation dataset. This dataset is used for the experimental evaluation of the proposed system. (Color adjusted for readability)

Figure 4: The distribution of plants in the examined dataset presenting the number of plants and their respected area measured in mm²

Figure 5: Examples of manually annotated images that were used in the training set. (Color adjusted for readability)

5. EXPERIMENTAL EVALUATION

The following experimental evaluation process reveals the contribution of individual parameters to the recognition yields of the proposed system. Moreover, it supports the claims made regarding the contributions of this
paper about the recognition performance of the Broad-leaved dock plant under challenging real world conditions.

Table 2: Description of image acquisition session and their individual parameters

<table>
<thead>
<tr>
<th>Location</th>
<th>Ancenis, France</th>
</tr>
</thead>
<tbody>
<tr>
<td>Farm type</td>
<td>Organic farm</td>
</tr>
<tr>
<td>Date</td>
<td>November 15, 2016</td>
</tr>
<tr>
<td>Weather</td>
<td>Cloudy</td>
</tr>
<tr>
<td>Canopy use</td>
<td>No</td>
</tr>
<tr>
<td>Velocity (m/s)</td>
<td>1.0</td>
</tr>
<tr>
<td>FPS</td>
<td>8</td>
</tr>
<tr>
<td>Start Time</td>
<td>16:56</td>
</tr>
<tr>
<td>Shutter (ms)</td>
<td>0.1</td>
</tr>
<tr>
<td>Acq. Images</td>
<td>11786</td>
</tr>
</tbody>
</table>

5.1. *Rumex-Ancenis Dataset*

While—as already discussed—weed detection in grasslands has started attracting increased attention, no publicly available dataset existed. Therefore, all experiments are conducted using the Rumex Ancenis dataset, introduced by the authors of this work in [13]. The dataset was acquired using the robotic platform shown in Fig. 1 in an open field of an organic dairy farm in Ancenis, France. Details of the image acquisition process are provided in Table 2.

The used vision system consists of a monochrome camera (PointGrey GS3-U3-23S6M-C) mounted on a rigid beam 1040 mm above the ground level, facing vertically downwards. The acquired grayscale images have a resolution of $1920 \times 1200$ pixels, with a ground resolution of $0.318 \times 0.266 \text{ mm}$. This translates to a ground area coverage of $61 \times 32 \text{ cm}$ per frame. The velocity of the robotic platform during operation was 1 m/s, while the frame-rate of the camera was set at 8 frames-per-second. The combination between robot velocity and frame rate provided a desired $2/3$ ground area overlap between successive captured frames.

As seen in Fig. 3, the dataset consists of images depicting the Broad-leaved dock in its surrounding environment, i.e. grasslands. The grassland
The acquisition took place on November 15, 2016 during a cloudy session after sunset (16:56), i.e., a preferable time of operation since the robot can avoid unwanted interaction with pasture animals.

The annotation of each image, as proposed in [10], made use of a 3×3 grid, defining 9 sub-regions in each image. Each one of them was manually labeled as “weed” if it contained weed plants, or “background” otherwise. During the manual annotation process, 2298 weed plants were found in a dataset consisting of 11786 images. Annotated image examples can be seen in Fig. 5. Weed plants were in various growing states, with covering areas from 1012.34mm$^2$ to 96,646.19mm$^2$. Henceforth, we denote our novel dataset Rumex-Ancenis. The distribution of weed plants and their growing states based on their area of coverage in the examined dataset, can be seen in Fig. 4.

5.2. Evaluation process

For the experimental evaluation of our proposed visual recognition system, we perform an extensive comparison between features extracted from CNNs [15, 16, 17, 18, 19] from the MatConvNet CNN toolbox$^2$. We consider the aforementioned CNN architectures based on their use in competing methodologies in weed recognition, e.g., a fine-tuned AlexNet [15] is used by the authors in [30] for a plant identification task. Moreover, for the proposed system we advocate architectures that provide state-of-the-art results in their corresponding fields, such as VGG-16 [17] in object recognition [35], as well as, Inception_v1 [18] and variances of ResNet [19] for image classification. The feature for each examined image, is extracted by intercepting each CNN before the softmax layer. This results in a 4096-dimensional feature vector from AlexNet [15], VGG-F [16], VGG-16 [17] convolutional networks, a 1024-dimensional feature vector from Inception_v1 [18] and a 2048-dimensional feature vector from ResNet-50 and ResNet-101 [19].

The resulting feature vectors from training image blocks are used to train linear classifiers with various kernels. The linear classifiers of [38] include a L2-regularized with L2-loss logistic regression model using primal computation (L2regLogReg), linear L2-regularized with L2-loss SVC using dual (L2regL2lossSVCd) and primal computation (L2regL2lossSVCp), linear L2-

$^2$All experiments were performed using MatConvNet version 1.0-beta25
regularized with L1-loss SVC using dual computation (L2regL1lossSVCd), a linear SVC by Crammer and Singer (Crammer-Singer SVC), linear L1-regularized with L2-loss SVC (L1regL2lossSVC), linear L1-regularized logistic regression model (L1regLogReg) and linear L2-regularized logistic regression model with dual computation (L2regLogRegd). All classifiers have a fixed cost equal to 10, while they are compared using the same training/testing image sets across all experiments. Moreover, for each classifier their recommended default tolerance rates were used. In detail, for L2regLogReg, L2regL2lossSVCd, L1regL2lossSVC and L1regLogReg the rate was set to 0.01. On the other hand, for L2regL2lossSVCp, L2regL1lossSVCd, Crammer-Singer SVC and L2regLogRegd the rate was 0.1. All other training parameters are set to the LibLinear default values. Our training set consists of 50% of all available image data from the examined acquisition session, resulting in 5,893 frames while the remaining are used for testing.

In order to treat equally all competing algorithms that structurally rely on large amounts of data, we create even more regions during the training stage of each examined weed recognition system. To achieve this, each image is not partitioned in 9 regions but in 144 regions using an overlapping sliding window approach—thus resulting to a $144 \times 5,893 = 848,592$ training images out of which 12,169 contained weed plants, while 836,423 did not, see Section 5.3. As described in Sec. 5.1, the examined dataset was extracted under real world conditions, hence weed plants have a sparse distribution within each field. The unconstrained use of available data can cause overfitting during system training, since the background class (N) is far more dominant than the class describing the Broad-leaved dock (P). We counteract this data imbalance and system overfitting for all competing methodologies by using a subset of the images describing the negative class. After an initial evaluation process, the final training ratio was set to 1 to 3, as this found to eliminate system overfitting while being a providing a good balance between autonomy and recognition yields, see Section 5.4. The additional computations required to process the data are not relevant for the system’s performance since training is done offline, before deploying any system to real-world conditions.

We consider the weed plant as positives (P) and other parts of an image, e.g grass, as negatives (N). All Tables present the mean accuracy values calculated from true-positives (TP), true-negatives (TN), false-positives (FP) and false-negatives (FN), using $\text{Acc} = (TP + TN)/(TP + FP + TN + FN)$. False positive rate (false alarm rate) and false negative rate (miss rate) are calculated by $FPR = FP/(FP + TN)$ and $FNR = FN/(FN + TP)$ respec-
Table 3: Mean accuracy, False-Positive Rate, False-Negative Rate, Precision and Recall (all as %) for Various Sizes of Non-overlapping Windows.

<table>
<thead>
<tr>
<th>Grid</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>3×3</td>
<td>77.7±1.4</td>
<td>22.4±1.5</td>
<td>21.5±2.5</td>
<td>3.3±0.2</td>
<td>78.5±2.5</td>
</tr>
<tr>
<td>4×4</td>
<td>77.1±1.1</td>
<td>22.9±1.1</td>
<td>28.7±1.6</td>
<td>3.3±0.1</td>
<td>71.3±1.6</td>
</tr>
<tr>
<td>5×5</td>
<td>76.8±0.6</td>
<td>23.2±0.7</td>
<td>26.6±0.9</td>
<td>3.4±0.1</td>
<td>73.4±0.8</td>
</tr>
<tr>
<td>6×6</td>
<td>78.2±0.5</td>
<td>21.7±0.5</td>
<td>28.9±1.2</td>
<td>3.3±0.1</td>
<td>71.0±1.2</td>
</tr>
<tr>
<td>7×7</td>
<td>77.3±0.5</td>
<td>22.6±0.5</td>
<td>29.3±0.8</td>
<td>3.0±0.1</td>
<td>70.7±0.8</td>
</tr>
<tr>
<td>8×8</td>
<td>77.4±0.5</td>
<td>22.5±0.5</td>
<td>29.6±0.5</td>
<td>2.9±0.1</td>
<td>70.4±0.5</td>
</tr>
</tbody>
</table>

respectively. Moreover, mean Precision is calculated using \(\text{Prec} = \frac{TP}{(TP + FP)}\), and mean Recall using \(\text{Rec} = \frac{TP}{(TP + FN)}\).

5.3. Optimization of image partitioning

We first examine the image partitioning parameters. For all experiments in Table 3 we use the feature vectors from AlexNet [15] combined with a L2regL2lossSVCd classifier. Each frame is partitioned in a number of blocks that allow the system to individually access the contents of that block. The number of blocks was chosen based on the given camera setup and typical Rumex plants sizes.

As seen in Table 3, finer grid sizes with more number of regions generally provide a good detection accuracy. However, these grid sizes generally result in more false-negatives and consequently to lower mean recall, thus adversely affecting the overall system accuracy in recognizing the desirable weed plant. This shows that in the examined application small blocks describing small parts of plants are not that discriminative in comparison with larger regions that can capture leafs as a whole or even a cluster of several overlapping leafs. The overall best detection results in this comparison is achieved when using 3×3 blocks per frame. The size of each block confirms that it is suitable to capture leafs or parts of the examined weed plant type. As a result, the system with the 3×3 grid yields the lowest false-negative percentage in our comparison while keeping the false-positive in a similar level to other systems with finer content partitioning.

Then, we investigate the effects of overlapping in comparison with non-overlapping grids. The grids of Section 5.3 can be considered as a non-overlapping sliding window method. Table 4 shows experiments performed on non-overlapping windows (NO) and the overlapping sliding window (O) in
Table 4: Mean accuracy, False Positive Rate, False Negative Rate, Precision and Recall (all as %) using Non-Overlapping (NO) or Overlapping (O) Window During Training (Tr) and Testing (Ts).

<table>
<thead>
<tr>
<th>Tr</th>
<th>Ts</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
<td>77.7±1.4</td>
<td>22.4±1.5</td>
<td>21.5±2.5</td>
<td>3.3±0.2</td>
<td>78.5±2.5</td>
</tr>
<tr>
<td>NO</td>
<td>O</td>
<td>62.4±3.5</td>
<td>37.8±3.6</td>
<td>18.7±2.3</td>
<td>2.2±0.2</td>
<td>81.4±2.3</td>
</tr>
<tr>
<td>O</td>
<td>NO</td>
<td>81.1±0.9</td>
<td>18.8±0.9</td>
<td>23.1±1.6</td>
<td>3.8±0.2</td>
<td>76.9±1.6</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
<td>79.0±0.6</td>
<td>20.9±0.6</td>
<td>24.9±0.8</td>
<td>3.5±0.1</td>
<td>75.1±0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Blocks</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>77.7±1.4</td>
<td>22.4±1.5</td>
<td>21.5±2.5</td>
<td>3.3±0.2</td>
<td>78.5±2.5</td>
</tr>
<tr>
<td>2 hor.</td>
<td>18</td>
<td>78.1±0.9</td>
<td>21.9±0.9</td>
<td>21.5±0.9</td>
<td>3.3±0.1</td>
<td>78.5±0.9</td>
</tr>
<tr>
<td>2 ver.</td>
<td>18</td>
<td>79.6±0.4</td>
<td>20.5±0.4</td>
<td>20.7±1.6</td>
<td>3.6±0.1</td>
<td>79.3±1.6</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>81.1±0.9</td>
<td>18.8±0.9</td>
<td>23.1±1.6</td>
<td>3.8±0.2</td>
<td>76.9±1.6</td>
</tr>
<tr>
<td>9</td>
<td>81</td>
<td>84.3±0.4</td>
<td>15.7±0.4</td>
<td>22.2±1.5</td>
<td>4.5±0.1</td>
<td>77.8±1.5</td>
</tr>
<tr>
<td>16</td>
<td>144</td>
<td>85.5±0.3</td>
<td>14.4±0.3</td>
<td>21.7±1.1</td>
<td>4.9±0.1</td>
<td>78.3±1.1</td>
</tr>
</tbody>
</table>

both directions resulting in 4 subsections from each block. In order to draw conclusions regarding the effects of those methods in training and testing, we present every possible combination in the top-performing grid sizes of the previous experiment.

As seen in the first part of Table 4, when non-overlapping blocks are used for training (Tr) and an overlapping sliding window for testing (Ts) then false-negative rates are decreased and recall has a slight increase, i.e, system was able to detect more weed plants. Using the overlapping sliding window for training clearly shows benefits for the overall performance since more data are available for training the proposed system. But when it comes to testing, non-overlapping blocks show several advantages over the overlapping blocks. An overlapping sliding window provides more opportunities for erroneous detection decisions since roughly the same block is examined multiple times. As a result, the system that uses non-overlapping blocks is outperforming the latter in every metric used in this comparison. Moreover, it is expected that the use of an overlapping sliding window during operation, will adversely affect computational time due to the increased amount of block proposals.

However, more combinations of overlapping sliding windows should be examined, since the additional computational complexity of additional data is not crucial during the training phase and can be compensated by improved
Table 5: Mean accuracy, False-Positive Rate, False-Negative Rate, Precision and Recall (all as %) using training set balancing or without any balancing (last row).

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 : 1</td>
<td>85.5±0.3</td>
<td>14.4±0.3</td>
<td>21.7±1.1</td>
<td>4.9±0.1</td>
<td>78.3±1.1</td>
</tr>
<tr>
<td>1 : 3</td>
<td>96.0±0.2</td>
<td>3.6±0.2</td>
<td>47.6±0.7</td>
<td>12.3±0.5</td>
<td>52.4±0.7</td>
</tr>
<tr>
<td>1 : 8</td>
<td>98.5±0.1</td>
<td>0.9±0.1</td>
<td>69.9±1.3</td>
<td>25.3±1.1</td>
<td>30.0±1.3</td>
</tr>
<tr>
<td>1 : 10</td>
<td>98.7±0.0</td>
<td>0.6±0.0</td>
<td>73.2±0.7</td>
<td>28.5±0.7</td>
<td>26.8±0.7</td>
</tr>
<tr>
<td>1 : 20</td>
<td>99.0±0.0</td>
<td>0.2±0.0</td>
<td>83.3±0.8</td>
<td>43.4±1.1</td>
<td>16.8±0.8</td>
</tr>
<tr>
<td>1 : 30</td>
<td>99.1±0.0</td>
<td>0.1±0.0</td>
<td>87.8±0.3</td>
<td>52.7±1.3</td>
<td>12.2±0.3</td>
</tr>
<tr>
<td>native ratio</td>
<td>99.1±0.0</td>
<td>0.0±0.0</td>
<td>92.0±0.0</td>
<td>63.5±0.0</td>
<td>8.00±0.0</td>
</tr>
<tr>
<td>no balancing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

detection results. The second part of Table 4 represents the training detection performance yields of overlapping blocks. Using 2 steps for each block, with vertical step being along the robot’s driving direction, both setups yield a slight improvement over the non-overlapping windows, but both of them are lacking in terms of detection accuracy. The last rows of Table 4 show that by decreasing the step of each window both horizontally and vertically—thus increasing the number of extracted blocks used for training—yield improved detection results while reducing false-positive and false-negative rates. As a result, in our following experiments we train each weed detection system with the highest number of steps, i.e, 4 horizontal and 4 vertical resulting in 16 steps per block.

5.4. Dealing with training overfitting of imbalanced dataset

As described in Section 5.2, the examined dataset was extracted under real world conditions, i.e, weed plants have a sparse distribution in each field. The unconstrained use of available data can cause overfitting during system training, since the background class (N) is more dominant than the class describing the Broad-leaved dock (P).

In our previous experiments, we counteract this data imbalance by applying a one to one ratio between classes thus using all available data for the positive class and a random subset of the same size from the negative class. Table 5 presents the experiments with various training data ratios and their effects on system’s detection performance. As in Section 5.3, experiments are performed using the aforementioned combination of AlexNet [15] for feature extraction and the L2regL2lossSVCd classifier. The one to one
 training ratio is shown to yield the best false-negative rates, i.e., the system is able to recognize the desired weed better than other systems. However, the false-positive rates are the highest in this comparison which, as discussed in Sec. 1 is very harmful for our specific scenario. Moreover, all training ratios above one to eight show that the system becomes overfitted with the negative class which may result in low false positive rates but it also yields increased false-negative rates, i.e., the system loses the ability to recognize the positive class. So, we consider that the one to three ration between classes for the examined dataset provides a more balanced detection performance.

5.5. Evaluation of CNN features and classifiers

Furthermore, we evaluate the discriminatory capabilities of features extracted using various deep CNN architectures and their combination with classifiers. As seen in Table 6 each one of the examined CNN-computed features is favored by different classifiers. The overall best detection results are not defined by a single metric, but rather a balance among them.

Table 6: Mean accuracy, False-Positive Rate, False-Negative Rate, Precision and Recall (all as %) using AlexNet, VGG-F, VGG-VD-16 and Inception_v1 CNN in Combination with Various Classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>AlexNet Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
<th>VGG-F Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogisticReg</td>
<td>96.1±0.2</td>
<td>3.4±0.2</td>
<td>47.6±0.7</td>
<td>12.3±0.5</td>
<td>52.4±0.7</td>
<td>96.1±0.2</td>
<td>3.2±0.2</td>
<td>50.8±0.7</td>
<td>12.9±0.5</td>
<td>49.7±0.5</td>
</tr>
<tr>
<td>LogisticSVC</td>
<td>95.4±0.4</td>
<td>4.8±0.8</td>
<td>14.3±0.7</td>
<td>14.3±0.7</td>
<td>51.9±0.6</td>
<td>95.9±0.7</td>
<td>4.6±0.7</td>
<td>50.0±0.8</td>
<td>20.4±0.9</td>
<td>37.1±0.8</td>
</tr>
<tr>
<td>LogisticSVCp</td>
<td>96.2±0.1</td>
<td>3.1±0.1</td>
<td>49.4±0.6</td>
<td>13.6±0.3</td>
<td>50.6±0.6</td>
<td>96.3±0.4</td>
<td>2.7±0.1</td>
<td>52.0±1.2</td>
<td>14.4±0.4</td>
<td>49.2±1.2</td>
</tr>
<tr>
<td>LogisticSVCd</td>
<td>96.4±0.1</td>
<td>3.1±0.1</td>
<td>49.4±0.6</td>
<td>13.6±0.3</td>
<td>50.6±0.6</td>
<td>96.3±0.4</td>
<td>2.7±0.1</td>
<td>52.0±1.2</td>
<td>14.4±0.4</td>
<td>49.2±1.2</td>
</tr>
<tr>
<td>Crammer-Singer SVC</td>
<td>96.2±0.1</td>
<td>3.4±0.1</td>
<td>49.8±0.8</td>
<td>12.8±0.5</td>
<td>51.2±0.8</td>
<td>96.4±0.3</td>
<td>3.2±0.2</td>
<td>49.6±1.1</td>
<td>11.7±1.0</td>
<td>49.1±1.1</td>
</tr>
<tr>
<td>LogisticSVC</td>
<td>96.2±0.1</td>
<td>3.4±0.1</td>
<td>49.8±0.8</td>
<td>12.8±0.5</td>
<td>51.2±0.8</td>
<td>96.4±0.3</td>
<td>3.2±0.2</td>
<td>49.6±1.1</td>
<td>11.7±1.0</td>
<td>49.1±1.1</td>
</tr>
<tr>
<td>LogisticReg</td>
<td>95.4±0.1</td>
<td>3.1±0.1</td>
<td>49.4±0.6</td>
<td>13.6±0.3</td>
<td>50.6±0.6</td>
<td>96.1±0.4</td>
<td>3.0±0.1</td>
<td>48.8±0.5</td>
<td>12.8±0.4</td>
<td>49.3±0.5</td>
</tr>
<tr>
<td>LogisticReg</td>
<td>95.4±0.1</td>
<td>3.1±0.1</td>
<td>49.4±0.6</td>
<td>13.6±0.3</td>
<td>50.6±0.6</td>
<td>96.1±0.4</td>
<td>3.0±0.1</td>
<td>48.8±0.5</td>
<td>12.8±0.4</td>
<td>49.3±0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogisticReg</td>
<td>93.4±0.2</td>
<td>6.4±0.2</td>
<td>25.7±1.5</td>
<td>10.0±0.2</td>
<td>74.3±1.3</td>
<td>96.1±0.4</td>
<td>3.6±0.1</td>
<td>50.5±0.7</td>
<td>15.5±0.9</td>
<td>69.5±0.7</td>
</tr>
<tr>
<td>LogisticSVC</td>
<td>93.1±1.1</td>
<td>6.7±1.1</td>
<td>28.3±2.6</td>
<td>9.3±1.2</td>
<td>71.7±2.6</td>
<td>96.2±1.7</td>
<td>3.9±1.8</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
<tr>
<td>LogisticSVCp</td>
<td>93.6±0.8</td>
<td>6.2±0.8</td>
<td>26.6±1.3</td>
<td>10.4±0.8</td>
<td>74.6±1.1</td>
<td>96.2±0.8</td>
<td>3.9±1.8</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
<tr>
<td>LogisticSVCd</td>
<td>93.4±0.9</td>
<td>6.4±0.9</td>
<td>25.7±1.5</td>
<td>10.0±0.2</td>
<td>74.3±1.3</td>
<td>96.1±0.4</td>
<td>3.6±0.1</td>
<td>50.5±0.7</td>
<td>15.5±0.9</td>
<td>69.5±0.7</td>
</tr>
<tr>
<td>Crammer-Singer SVC</td>
<td>92.8±0.2</td>
<td>7.0±0.2</td>
<td>27.8±1.4</td>
<td>9.8±0.3</td>
<td>73.2±1.4</td>
<td>96.2±1.7</td>
<td>3.9±1.7</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
<tr>
<td>LogisticSVC</td>
<td>93.0±0.2</td>
<td>6.8±0.2</td>
<td>26.3±0.2</td>
<td>9.1±0.2</td>
<td>72.3±0.8</td>
<td>96.2±1.7</td>
<td>3.9±1.7</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
<tr>
<td>LogisticReg</td>
<td>93.4±0.3</td>
<td>6.9±0.3</td>
<td>26.3±0.3</td>
<td>9.1±0.2</td>
<td>72.3±0.8</td>
<td>96.2±1.7</td>
<td>3.9±1.7</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
<tr>
<td>LogisticReg</td>
<td>93.4±0.3</td>
<td>6.9±0.3</td>
<td>26.3±0.3</td>
<td>9.1±0.2</td>
<td>72.3±0.8</td>
<td>96.2±1.7</td>
<td>3.9±1.7</td>
<td>51.7±2.7</td>
<td>20.0±5.0</td>
<td>63.4±7.3</td>
</tr>
</tbody>
</table>

**Table 6**: Mean accuracy, False-Positive Rate, False-Negative Rate, Precision and Recall (all as %) using AlexNet, VGG-F, VGG-VD-16 and Inception_v1 CNN in Combination with Various Classifiers.

- **Classifier**: Model used for classification.
- **Accuracy**: Mean accuracy of the model in percentage.
- **FPR**: False-Positive Rate in percentage.
- **FNR**: False-Negative Rate in percentage.
- **Precision**: Precision of the model in percentage.
- **Recall**: Recall of the model in percentage.
Figure 6: Comparison between the best combinations for each examined CNN model. Left: ROC curve for Broad-leaved dock (Positive class). Middle: ROC curve for background (Negative class). Right: Precision-Recall curve.

The CNN architecture of [15] yields its best and most balanced detection results in combination with the L2regLogReg classifier. The combination of L2regL2lossSVCp with the CNN architecture of [16] yields both the best accuracy and false-positive rate when compared to other classifiers using the same model. In conjunction with other metrics this combination yields the best performing weed detection system for the examined CNN model. When using the VGG-16 CNN model [17], the most balanced performance is achieved using L1regLogReg classifier. Compared to other classifiers using the same features, the L1regLogReg classifier yields the best detection accuracy, retains a low false positive rate while keeping a good balance between Precision and Recall. The Inception CNN [18] provides its best detection results when combined with the L2regLogReg classifier. The combination of this examined architecture with Crammer-Singer SVC classifier may seems as if it is outperforming the previously mentioned combination. However, a closer look at the standard deviation of false-negative rates and Recall reveals that this combination provide more unstable and thus unreliable detection results.

Features extracted with ResNet-50 CNN [19] show its best overall result when combined with the L2regLogReg. This classifier may seem slightly out-performed by L1regLogReg, but it achieves more stable results as shown by the standard deviation in all metrics. Finally, the best overall combination of features extracted with ResNet-101 [19] are achieved using the L2regLogReg. Similar, to its “shallower” counterpart the overall system performance is better than the one using L1regLogReg due to less deviation.

Using the best combinations for each examined CNN model, we perform
a comparison of their yielded detection performance; these results are illustrated in Fig. 6. The VGG-16 [17] achieves the least false-positive rate and the highest Recall in our experimental comparison, i.e., the best detection of Broad-leaved dock plants. However, it is the least effective model in this comparison due to the increased false-positive rate. This results in a decreased system accuracy. The VGG-F CNN model [16] achieved the best detection accuracy and the best false-positive rates in this comparison. The fact that it produces higher false-negative rates and high standard deviation in Recall shows a very unstable system in terms of weed detection. The CNN model in [15] provides a more balanced detection performance in terms of accuracy, Precision and Recall than the aforementioned architectures with its main disadvantage being its increased false-negative rates. The combination using the Inception CNN combined with the L2regLogReg logistic regression classifier achieves the best detection performance, yielding a good detection accuracy and low false-positive and false-negative rates.

Finally, both ResNet-50 and ResNet-101 [19] achieve overall similar results, but the “shallower” ResNet-50 network provides better FPR in this comparison. On the other hand, its increased FNR shows that the features are not as descriptive for the positive class as the ones from Inception. Therefore, Inception CNN with the L2regLogReg logistic regression is the combination we adopt for our proposed system and will be used for all our further experiments.

5.6. Comparison with other weed detection methodologies

We compare the proposed system to other competing methodologies on the examined dataset. The chosen rival methods were selected according to their relevance and applicability in our scenario—weed recognition in grasslands—as discussed in Sec. 2. For this comparison, we respect the settings of each methodology while implementing them in a similar training/testing setup to match the one used for our proposed weed recognition system. As a result, all algorithms are using the same image data. We should note that for every experiment, accuracy and recall measures are affected by the recognition of the negative class due to the largely uneven ratio between the negative and positive class. Due to the requirements of the examined weed control problem, as described in Sec 1, false-positive rates greatly affect the robotic weed control system’s autonomy. Hence, we target a trade-off between false-negative rate and precision, i.e., the system’s ability to recognize weed plants in grasslands.
Table 7 presents the results obtained by all competing methodologies \[10, 11, 12, 21, 30\] in comparison with the new proposed weed recognition system. We start our experimental comparison with the conventional hand crafted feature-based methods \[10, 11, 12\], we continue with deep learning-based methodologies \[21, 30\] and conclude with the proposed systems.

The SIFT-based system of \[10\] provides the lowest accuracy from all examined methods. This is partly due to the system’s high false-positive rates, i.e., its inability to efficiently recognize the negative class. While this system (\[10\]) yields great recognition performance of the positive class, its inability to describe the background class (N) makes it unsuitable for real-world implementation. The SURF-based system in \[12\] is also following the same trend. The examined system slightly outperforms the recognition capabilities of \[10\], with improvements in false-positive and false-negative rates. However, the system is still affected by the high false-positive rates that hinder the autonomy of a weed control robot.

Table 7: Mean accuracy, False-Positive Rate, False-Negative Rate, Precision and Recall (all as %) Comparison Between Weed Recognition Methodologies.

<table>
<thead>
<tr>
<th>Method description</th>
<th>ref.</th>
<th>Accuracy</th>
<th>FPR</th>
<th>FNR</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT feature based system</td>
<td>10</td>
<td>86.5±0.8</td>
<td>13.3±0.8</td>
<td>35.1±1.4</td>
<td>4.5±0.3</td>
<td>64.9±1.4</td>
</tr>
<tr>
<td>SURF feature based system</td>
<td>12</td>
<td>88.2±0.9</td>
<td>11.6±0.9</td>
<td>33.6±2.1</td>
<td>5.2±0.4</td>
<td>66.4±2.1</td>
</tr>
</tbody>
</table>

From the above, we can conclude that even though hand crafted feature-based methods provide a good representation of the positive class—which even outperforms some of the examined deep learning-based methodologies—they are still lacking in correctly detecting the negative class due to the inefficient discriminatory capabilities of their feature vectors.

The deep learning methodology in \[21\] is similar in architecture to the proposed systems, using the Overfeat CNN architecture \[28\] for feature extraction and using Extremely randomized trees \[27\] for classification. As
Figure 7: Comparison between the best performing proposed system (Inception_v1 + L2regLogReg) and competing methods of Table 7. Left: ROC curve for Broad-leaved dock (Positive class). Middle: ROC curve for background (Negative class). Right: Precision-Recall curve. In the middle and left plots the results of [10] are not visible as they are overlapped by the results of [12].

shown in Table 7 that system yields increased recognition for the class found in majority, i.e., the negative class, while achieving increased accuracy and the lowest false-positive rate for the examined dataset. However, the overall performance of this system is fictitious since the false-negative rates are the highest in this experimental comparison. So, the system is able to learn extremely well the negative class but is not able to recognize weed plants.

As described in Sec. 2, the method in [30] performs fine-tuning of the AlexNet CNN architecture [15] for plant recognition. We consider the same training parameters as the ones used in [30]. More specifically, the learning rate for the softmax layer is set to 10, the learning rate for all the other layers was set to 0.1 and the network was optimized using stochastic gradient descent (SGD) for 50000 iterations. In this comparison, the method in [30] is fine-tuned using the proposed Rumex-Ancenis dataset where it is shown to provide a good overall accuracy with small FPR, which also affects the recall rates yields of this method. However, FNR shows that the fine-tuned network does not provide good recognition performance for the positive class, i.e., weed plants. This is mostly due to the tendency of fine-tuned CNNs to overfit, especially when trained with a small amount of data.

The new methodologies we are considering and testing in this work use the well-known CNN architectures [15, 16, 17, 18] in order to extract features that are then classified using the logistic regressions and SVMs of [38], as presented in Sec. 5.2. All resulting combinations between the selected CNN architectures and classifiers are 32, derived from 8 classifiers and 4 CNN architectures. However, for the sake of an efficient presentation we chose to
present only the best performing combination from each of the examined CNN architectures. We evaluate the discriminatory capabilities of features extracted using various deep CNN architectures and their combination with classifiers. As seen in Table 7, each one of the examined CNN-computed features is favored by different classifiers. As before, the overall best recognition results are not defined by a single metric, but rather a balance among them.

The CNN architecture of [15] in combination with the L2regLogReg classifier yields its best and most balanced recognition results in terms of accuracy, precision and recall than other architectures, its main disadvantage being its increased false-negative rates.

The combination of L2regL2lossSVCp with the VGG-F CNN architecture [16] yields the best recognition accuracy and the best false-positive rates from all the new systems proposed in this paper. However, the fact that it produces higher false-negative rates and high standard deviation in recall shows a very unstable system in terms of weed recognition.

When using the VGG-16 CNN model [17], the most balanced performance is achieved using L1regLogReg classifier. This system achieves the least false-positive rate and the highest recall in our experimental comparison, i.e., the best recognition of Broad-leaved dock plants. However, it is the least effective model in comparison to other deep learning-based methods due to the increased false-positive rate, which results in a decreased system accuracy.

Finally, the Inception CNN [18] provides its best recognition results when combined with the L2regLogReg classifier that achieves the best recognition performance, yielding a good recognition accuracy and low false-positive and false-negative rates.

Comparing the performance of classifiers, we should also note that logistic regression classifiers have shown a better adaptation to CNN-based feature extraction methods, thus outperforming the linear SVMs, as well as the extreme randomized tree classifier of [21] in terms of classification accuracy.

Due to the discriminatory capabilities of Inception based features and the efficiency of the L2regLogReg logistic regression classifier, this system is clearly shown to outperform all proposed systems and competing methodologies, as shown by the ROC and Precision-Recall curves in Fig. 7.

Fig. 8 illustrates the recognition performance of our proposed system with respect to the plants’ projected ground area. The results show that the majority of plants larger than 20000 mm$^2$ were correctly recognized from our system.
Figure 8: Histogram of the projected ground area for correctly detected (blue) and missed (yellow) plants in the test set for the plant detection case.

Figure 9: Example frame of results using the proposed weed detection system. Red boxes indicate false-negative results, yellow boxes indicate false-positive results and green boxes indicate true positive detection results. Unlabeled areas are depicting true negative results. (Image was processed using histogram equalization for demonstration purposes).

6. CONCLUSIONS

We proposed a system for the detection of the Broad-leaved dock in grasslands. The output of our developed system is visualized in Fig. 9. The problem is difficult to solve using just high resolution 2D images. However, the analysis of relevant literature shows that a promising approach would be to apply transfer learning in order to exploit the strengths of pre-trained CNNs. The discriminative power of the features generated by the CNNs in conjunction with robust classification techniques yield very promising results, outperforming previous hand crafted feature-based and deep learning-based methods.
The main drive when developing our system was to combine high accuracy with the lowest possible FPR. These would ensure both effective and efficient operation of the robotic weeding system, since it would not waste time and resources on dealing with non-existent weeds—as would be the case for high FPR. The results of this work on a dataset captured under real conditions are very encouraging and show that the problem can be sufficiently addressed even with the mere use of 2D vision.

For future work, several methods that greatly improve the overall system performance can be considered, to achieve lower FNR and FPR. First, our current dataset could be expanded with additional data captured under different conditions, i.e., illumination, grass density, etc. This will potentially lead to more robust weed recognition, and more effective fine-tuning and training of deep learning-based methodologies without overfitting. Since the distribution of the positive class is so sparse in this examined weed detection problem, future work could benefit from synthetic sampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN). Furthermore, novel computationally efficient architectures, such as Xception and MobileNetv2, should also be considered since they will allow for more computationally efficient implementations. Moreover, we advocate the use of online training approaches, that would allow the system to learn from small datasets and continue learning while in operation. Finally, the use of pixel-wise segmentation, instead of the current region-based one, and detection methods inspired by novel object detection algorithms can additionally allow the detection of plants, as well as individual parts of them, e.g., stems, as required for weed control.

7. Acknowledgments

This work has been supported by the DockWeeder project (project ID: 30079), administered through the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618123 [ICT-AGRI 2]. The project has received funding from the Ministry of Economic Affairs (The Netherlands), from the Federal Office for Agriculture (Switzerland), and from Innovation Fund Denmark, the Ministry of Science, Innovation and Higher Education (Denmark).
References


DockWeeder, The dockweeder robot enables organic dairy farming by controlling grassland. European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618123 [ICT-AGRI 2].
URL http://dockweeder.eu/


URL http://arxiv.org/abs/1409.1556

URL http://arxiv.org/abs/1409.4842

URL https://doi.org/10.1109/CVPR.2016.90


URL https://doi.org/10.1007/s10994-006-6226-1

URL http://arxiv.org/abs/1312.6229

URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848


URL http://arxiv.org/abs/1804.02767

31
URL http://arxiv.org/abs/1506.01497


URL http://doi.acm.org/10.1145/2733373.2807412


URL https://doi.org/10.1109/CVPR.2017.195