Weed Detection in UAV Images of Cereal Crops with Instance Segmentation

Master Thesis in Data Science

Arina Gromova

Supervisor: Dr. Hamam Mokayed, Postdoctor at LTU

External Supervisor: Igor Tihonov, CEO at Solvi AB

October 12, 2021

Abstract

Modern weeding is predominantly carried out by spraying whole fields with toxic pesticides, a process that accomplishes the main goal of eliminating weeds, but at a cost of the local environment. Weed management systems based on AI solutions enable more targeted actions, such as site-specific spraying, which is essential in reducing the need for chemicals. To introduce sustainable weeding in Swedish farmlands, we propose implementing a state-of-the-art Deep Learning (DL) algorithm capable of instance segmentation for remote sensing of weeds, before coupling an automated sprayer vehicle. Cereals have been chosen as the target crop in this study as they are among the most commonly cultivated plants in Northern Europe. We used Unmanned Aerial Vehicles (UAV) to capture images from several fields and trained a Mask R-CNN computer vision framework to accurately recognize and localize unique instances of weeds among plants. Moreover, we evaluated three different backbones (ResNet-50, ResNet101, ResNeXt-101) pre-trained on the MS COCO dataset and through transfer learning tuned the model towards our classification task. Some well-reported limitations in building an accurate model include occlusion among instances as well as the high similarity between weeds and crops. Our system handles these challenges fairly well. We achieved a precision of 0.82, recall of 0.61, and F_1 score of 0.70. Still, improvements can be made in data preparation and pre-processing to further improve the recall rate. All and all, the main outcome of this study is the system pipeline which, together with post-processing using geographical field coordinates, could serve as a detector for half of the weeds in an end-to-end weed removal system.

Contents

1 Introduction

Agriculture worldwide is facing the challenge of increasing food production to sustain the growing population [1]. A large portion of the losses in harvest is due to weeds out-competing cultivated plants, consuming nutrients and sunlight at every stage of growth [2, 3]. The main method for weed control in many parts of the world, including Sweden, is spraying fields on ground level with chemical pesticides [2, 4]. Whole areas are usually covered without taking the weed status in the local environment into consideration. Chemical pesticides mainly consist of herbicides that cause toxicity and contamination in nature [4]. Their increased use imposes a threat to human health, losses in biodiversity, and changes the local ecosystem [5]. In response, current research within precision agriculture aims to incorporate AI solutions for site-specific weed management to reduce the need for herbicides [4, 6]. More specifically, Convolutional Neural Network (CNN)-based methods are making the greatest advances in remote sensing of weeds prior to taking early and accurate actions [3, 6, 7]. This work touches precision agriculture for weed management targeting cereal crops in Swedish farmlands.

1.1 Background

Following the UN goals to achieve sustainable and safe food production [1], weed management is evolving towards becoming data-driven with automated weeding robots that can minimize the use of herbicides. Robotic weeding commonly involves site-specific spraying from a programmed vehicle reducing the volumes that are needed, compared to traditional spraying [4]. Even solutions based on selective mechanical, electrical, or thermal weeding are mentioned in the literature, in which case the need for chemicals would be eliminated [3, 8]. As a result, automated weed management systems contribute to saving the farmer costs from pesticides and fuel, increasing harvest quality, as well as improving the status of the surrounding nature [2, 3].

Precision agriculture is a growing field where data serves as a basis for decision support concerning cultivation and harvest [9, 10]. For weed management, the procedure spans from collecting data of the current weed status in the field, to remote sensing (including detection of single weeds, calculating growth density, and identification of species), and finally, to building a steering system for the weeding vehicle using geographical metadata [2, 4, 11].

Visual imagery is preferred to use for weed detection as it can carry information on vegetation patterns, growing conditions, as well as physiological and morphological features of plants [7]. Images can be collected through cameras mounted on All-terrain Vehicles (ATV), or more advantageously, on Unmanned Aerial Vehicles (UAV). UAVs, or drones, enable fast acquisition of images at desired height, can be remotely controlled, and provide high pixel resolution at a low cost [12]. A high-resolved map of the whole field can be constructed by flying in a predetermined pattern followed by overlaying the output images [4]. A section of a field map showing farmed cereal line crops in Sweden can be seen in Figure 1. A single UAV image of the cereal field can be seen in Figure 2. In Figure 3, further zooming into the image shows highly resolved instances of young crops with weeds growing in-between.

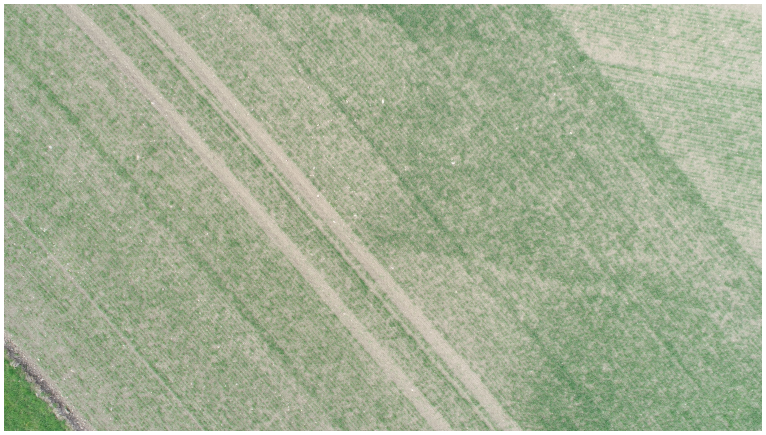


Figure 1: Section of a high-resolution field map made from UAV images. The map shows cereal crops cultivated in Sweden. Image provided by Solvi [13].

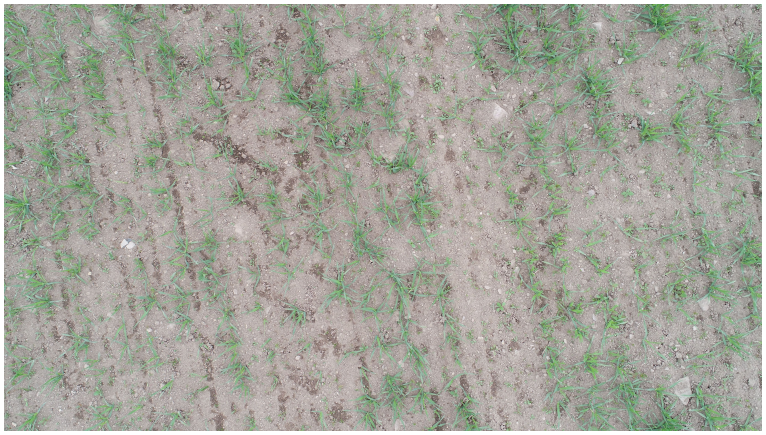


Figure 2: UAV image with 5472 × 3078 pixel resolution of cereal plants, provided by Solvi [13]. Imagery used for training the CNN model after data preparation.

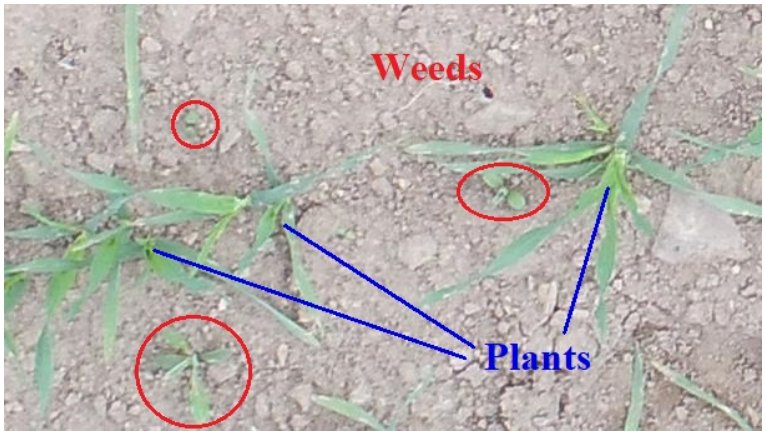


Figure 3: A 608 × 342 pixel crop out of a UAV image capturing young cereal plants with small weeds growing in-between.

Identifying and locating all weeds in the field is a challenging task, and creates a bottleneck in the development process [2, 3]. The most critical step is accurate recognition, as weeds can have very similar shape and color to cultivated plants [3, 7, 9]. Miss-classification may lead to inefficient actions or can even harm the crop [2]. Therefore, large research efforts have been focused on developing state-of-the-art algorithms that can process visual information to locate and classify weeds from plants [6, 11, 12]. Until recently, no Machine Learning (ML) approaches were able to provide an end-to-end solution.

The entrance of Deep Learning (DL) frameworks using CNN as a backbone have gained a lot of attention as they show the most promising results in computer vision tasks, much thanks to the development of sensors, the increase in computational power (especially Graphical Processing Units, GPUs), and to open source initiatives [3, 7, 9, 12, 14]. For object detection, different pre-trained CNNs exist that can be fine tuned to learn a new task. While some available networks perform object localization by drawing an identification box (bounding box, or bbox) around the class object, instance segmentation enables pixel-wise identification of each instance [15]. Due to the complex shape of plants, instance segmentation can be more suitable for remote sensing of weeds [8, 7, 16, 17] and has therefore been chosen as the main approach in this study.

1.2 Motivation

Nationally transitioning towards more sustainable solutions in Swedish farming is a responsibility of the Swedish Board of Agriculture (Jordbruksverket). Jordbruksverket sets the Swedish Plant Protection Strategy [18] and provide governmental funds to research contributing to reach its goals. Supported by Jordbruksverket, a large collaborative project in site-specific weed management has been initiated by Agroväst [19] - an organization that specializes in introducing front-end technologies into Swedish agriculture. Together with one of the key collaborator companies, Solvi [13] - a startup that specializes in drone-based crop monitoring using AI, the thesis aims to train a CNN-based DL framework for a future weed control system to localize single weeds among cereal crops. Cereals such as winter wheat, rye and barley make up around half of the cultivated crops in Northern Europe [20] and are thus an important target. Conclusively, our objective is to contribute to sustainable Swedish agriculture.

1.3 Problem Description

Remote sensing of weeds is a common problem covered in the literature. Authors have reported outcomes of many well performing models tested in their specific experimental settings. Yet, the solution is different each time new geographical locations and species of crops or weeds are introduced. In their review, Hasan *et al.* [11] list 70 publicly available annotated datasets that have been generated from weed detection studies, the majority targeting crop such as soybean, sugar beet, carrot, or maize. A handful of datasets include young barley, wheat or rye, but are either too small, too specific, or lack instance annotations (binary masks, or polygon drawings).

Solvi has collected UAV images of cereal fields in the Southern and Western Coast regions of Sweden to use for training a weed detector. The farms had permitted to use their data for this project. Example images from one location can be seen in Figure 1 - 3. A smaller portion of the large image dataset has been annotated at pixel level by experts in precision agriculture before the start of this study. To train an instance segmentation model, we propose implementing a Mask Region-based Convolutional Neural Network (Mask R-CNN), pre-trained on the benchmark MS COCO dataset [21]. Detectron2 [22] - an open-source software platform was chosen for implementation of the algorithm. Detectron2 comes with a set of pre-built computer vision frameworks, including Mask R-CNN.

1.4 Challenges

As previously stated, detection of weeds growing among plants is challenging. At times, weeds can easily be separated from line crops by space when weeds opportunistically grow in between the planted rows [2]. Very often, occlusion is present, which has been brought up as an obstacle in similar studies [3, 20]. Furthermore, weeds and crops have similar features, such as shape and color, and can be hard even for the human eye to distinguish. The morphology, texture and spectral properties change as the plant grows which introduces uncertainties to the classification. Finally, lightning conditions, shadows, and motion blur in captured aerial or on-ground images contribute to lower quality of input data [2, 11]. On the other hand, only including few high-quality images where classes are easily identified, may produce well performing models during the evaluation procedure, but will certainly have limitations when deployed in the field.

Implementation of deeper and more advanced models to enable learning of difficult visual tasks come with increased numbers of parameters to optimize [7]. This contributes to increased training time and consumption of computational resources, which might easily expand over the scope of this project. In addition to that, large quantities of data are generally required to train deep networks to reduce the risk over-fitting [12, 7] This requirement is ever so challenging to meet due to the time consuming annotation procedure of images, where each plant in the dataset needs to be drawn manually. Including objects without ground truth annotations falsely penalizes the model during training and must be avoided.

Another limitation is selecting the framework to implement. Tuning the problem towards instance segmentation limits the candidate pool of available CNN-based architectures and backbones. The choice can have a great impact on the training time but also lead to different results. All backbones and CNN methods cannot be tested here, so we have chosen the DL framework Mask R-CNN that is reported to achieve one of the highest performances on benchmark datasets, online data science challenges, and in published studies, while at the same time being rather fast [15].

1.5 Project Boundaries

In an attempt to meet the challenges described within given scope of this project, we have set following boundaries.

- Use the UAV dataset that has been produced by Solvi, including plain RBG images in JPG format as well as TIFF images that include geographical metadata.
- Convert the available annotations to a format readable by the Detectron2 platform instead of producing new.
- Investigate Mask R-CNN for instance segmentation.
- Trial a few CNN backbones and compare their performances.
- Due to time restrictions, skip pre-processing and augmentation of images. Instead, discuss those approaches for future work.

1.6 Thesis Structure

The rest of this thesis is organized as follows. An overview of the theory behind Deep Learning approaches for computer vision is laid out in **Section 2**, covering neural networks for classification, convolutional neural networks with images as input, Region-based CNNs and Mask R-CNN, available frameworks for transfer learning, and finally, performance metrics for evaluation of object detection models. In **Section 3**, related work in DL-based weed detection is presented and a few relevant publications are covered in more detail. Methodology of this study is explained in **Section 4**, covering data acquisition, data preparation, model training and evaluation, as well as presenting an overview of the system architecture. The results are presented in **Section 5**, followed by a discussion in **Section 6**. Finally, our conclusion and suggestions for future work are covered in **Section 7**.

2 Theory

Compared to traditional Machine Learning, Neural approaches (the domain where Deep Learning and Convolutional Neural Networks belong) do not require manual feature extraction as they possess the ability of end-to-end learning [7]. Algorithms that work in this manner have proven to perform with higher accuracy in machine vision tasks [3].

2.1 Neural Networks

Neural networks are algorithms that mimic the way neurons in the brain process information to classify a sample based on its features [23]. A general network structure is illustrated in Figure 4. The simplest networks consist of just a few nodes in a hidden layer that each applies a non-linear activation function to the input data [7]. The activation function decides whether the node is active to pass on the signal or

not. One very common activation function for nodes is ReLu, described in Equation 1. Softmax, described in Equation 2, is commonly used in the output layer node, or nodes, to decide the class belonging. The nodes in the input layer, hidden layer, and output layer are connected through weights and biases through which the data is propagated and transformed to match the output class. Learning takes place during backpropagation when an optimization algorithm (such as gradient descent) is applied to adjust the network weights by calculating and minimizing the training loss. For each batch of data, the network parameters are updated and after one epoch, all training data has been run through the network. The total number of training iterations is therefore proportional to epochs, number of samples, and batch size, according to Equation 3.

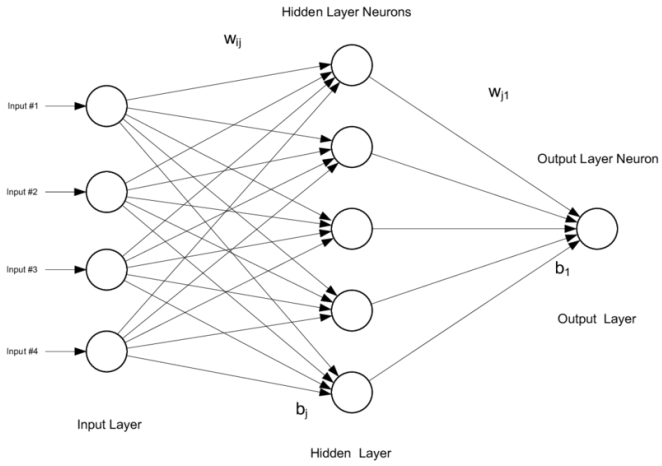


Figure 4: Basic Neural Network structure illustrated by Alam [24].

$$ReLu(x) = max(0, x) \tag{1}$$

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{class_n} e^{x_j}} \tag{2}$$

$$Iterations = \frac{Epochs \times N_{observations}}{Batch\ size} \tag{3}$$

Increasing the number of hidden layers and transformations in the network architecture adds to the complexity and computational requirement during training. Depending on the problem at hand, the architecture can be set up with additional structures. Deeper networks are capable of extracting higher-level features to solve more abstract problems, such as learning spatial patterns in images [7]. CNN's have been designed specifically for such tasks.

2.2 Convolutional Neural Networks

A Convolutional Neural Network consists of layers with nodes that perform convolutions on data tensors (higher dimensional data matrices such as RGB pixel values of images) by applying filters in a sliding window approach [7]. A general overview of the CNN architecture is illustrated in Figure 5. Spatial features such as corners, textures, edges, and shapes, are extracted from the data structure by calculating the dot product between the filter and pixel values. The transformations produce a progressively abstract feature map that is successively propagated to deeper parts of the network. During the backpropagation the randomly initiated filter values are optimized to extract features from the images to match the class outcome. Pooling layers are added in the CNN structure to down-scale the large feature space which helps exaggerate data structures, reduce the computational cost, and prevent over-fitting. The most common example is Max-Pooling, where the largest value in every window frame is extracted. In the final classification step in the network, flattening and dense layers are added. A classification function, such as Softmax, predicts the class output from the final feature vector.

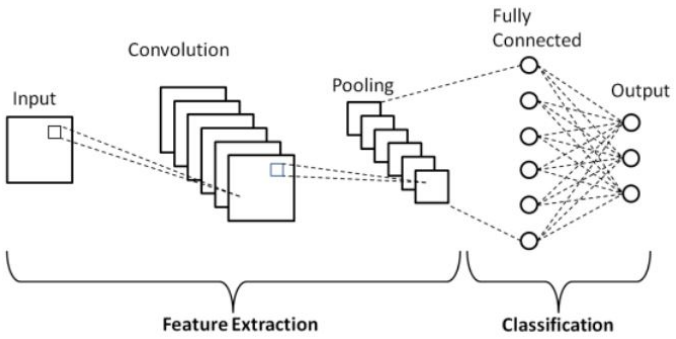


Figure 5: Simplified overview of the CNN architecture, illustrated by Phung *et al.* [25]. The structure generally consists of two basic parts - feature extraction and classification [11]. Several convolutions and pooling layers can be added in a design that suits the problem at hand.

Implementing CNNs for image classification has been successful in the past, but limits the output to assign one class to the whole image [26]. To be able to handle classification and localization of more than one object within an image, new methods have been produced with additional structures built on top of the CNN backbone.

2.3 Region-based Convolutional Neural Networks

To enable detection and localization of objects of different classes using bounding boxes (bboxes), a method that uses Regions with CNN features has been developed by Girshick *et al.* [26]. The system architecture for Region-based CNN (R-CNN) is set up in three modules; first applying a selective search algorithm to extract around 2000 smaller, category-independent regions within the input image, secondly, applying a backbone CNN to extract a feature vector, and thirdly, performing category-specific classification using linear Support Vector Machine (SVM). Image warping using mean

subtraction is added in the region proposal step to create a fixed-size (227 × 227 pixels) matrix of the extracted region to be used as classification input. The architecture overview is presented in Figure 6.

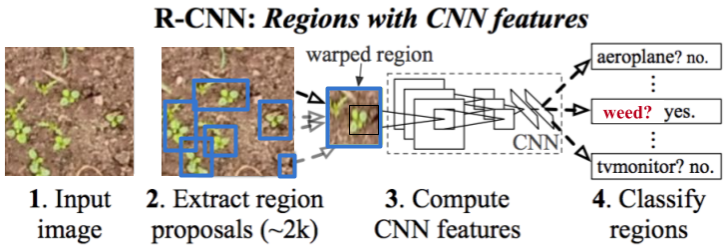


Figure 6: Overview of the R-CNN architecture, based on the illustration made by Girshick *et al.* [26].

R-CNN has been evolved further to Fast R-CNN [27] and Faster R-CNN [28], improving accuracy (when tested on benchmark datasets such as PASCAL VOC and MS COCO). Moreover, the computation speed has been improved by refining the region proposal step and enabling parallel computation on the GPU. Structures of the Faster R-CNN are demonstrated and compared to Mask R-CNN in Figure 7.

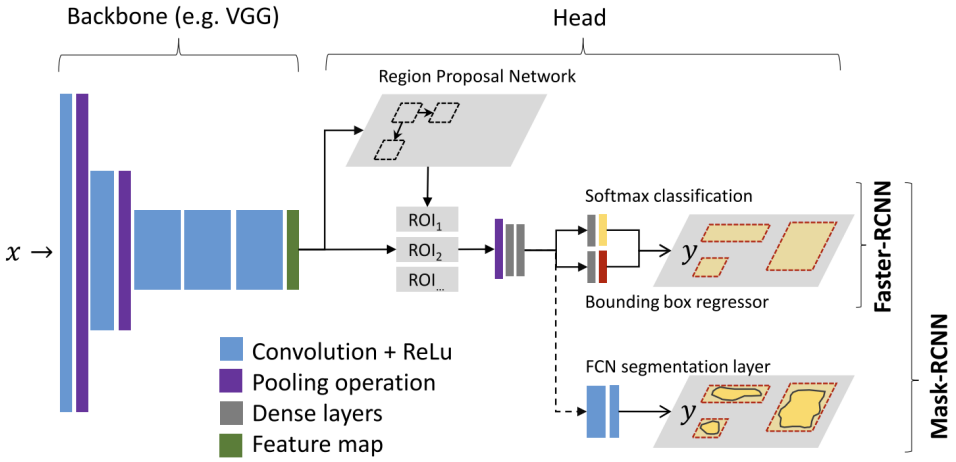


Figure 7: Network structure of Faster R-CNN and derivation of Mask R-CNN instance segmentation illustrated by Kattenborn *et al.* [7].

In the updated method pipeline of Fast and Faster R-CNN, a Region of Interest (RoI) proposal branch has been introduced and the classification algorithm exchanged. First, a CNN network is used to generate a convolved feature map of the input image from which RoIs are extracted by pooling operations [27]. This part constructs the backbone architecture. Object detection is executed at the head part of the network where two parallel processes are implemented to generate outputs; one predicting the class label of the object within the RoI by a Softmax classifier, and a second providing object localization through a bounding box regressor. In the latest version of Faster R-CNN, a Region Proposal Network (RPN), which is trained additionally on each specific

task, is added to fine tune the region proposal procedure [28]. RPN produces rough estimations of object locations from the feature map, called anchors, that fine-tune the candidate RoIs [29].

Mask R-CNN [15] is an extension of Faster R-CNN to enable pixel-wise segmentation of the object using instance masks. Masks are binary pixel drawings of each unique object to separate it from its background and other instances. Masks are used to train an added segmentation branch in the head classification part containing a Fully Convolutional Network (FCN) while bounding boxes surrounding the object are used for bbox regression [7]. The FCN thus adds a segmentation prediction to the RoI output.

2.4 Other Deep Learning frameworks for Object Detection

Apart from the R-CNN family of DL frameworks for object detection, there is also the fully connected You Only Look Once (YOLO) [30] and Single Shot Detector (SSD) [31] families. SDD includes some elements similar to Faster R-CNN such as creating feature maps and multi-scale region proposals but is faster to compute [20]. YOLO has much faster training and prediction speed as it can see the whole image at once, which is often more suitable for real-time deployment [29]. The drawbacks of SDD and YOLO are lower accuracy compared to R-CNN when tested on benchmark datasets, but more importantly, there is no implementation of pixel-wise segmentation.

2.5 CNN Backbones

Baseline CNN models have already been trained on large generic image datasets (such as PASCAL VOC, ImageNet, or MS COCO that contain thousands of images on e.g. vehicles, people, animals, and buildings), and can be used for transfer learning on unseen data to further tune the model weights [7, 11]. Implementing a pre-trained backbone within the DL framework for feature extraction will therefore drastically reduce the number of training iterations, while still preserving a high accuracy. VGGNet (commonly VGG16 or VGG19), ResNet (commonly ResNet50 or ResNet101), AlexNet, GoogleLeNet, Inception, or DetectNet, are among the most popular CNN backbones for weed detection [11, 16, 29].

Deep Residual Nets (or ResNet) [32], is commonly used as the backbone within Mask R-CNN instance segmentation tasks in the literature and the architecture can be designed in different structural configurations. The most applied versions of ResNet use 50 or 101 convolutional layers, organized within five modules (C1 to C5). A recent addition to the baseline models is the ResNeXt backbone [33] which currently runs 32 blocks of ResNet in parallel, consuming more computation time but achieving improved accuracy in some studies [15, 34].

To upscale the feature map in order to recognize objects of different sizes, different configurations can be implemented on top of ResNet. For example, early stopping using the C4 output, or dilated (multiplied with a constant) C5 output are available [15]. The most popular configuration uses Feature Pyramid Network (FPN) [35] to expand the C5 feature map output in a top-down pyramid-like manner.

2.6 Performance Metrics

In object detection and instance segmentation, correct classification is determined by evaluating prediction overlap with the ground truth by calculating intersection over union (IoU), described in Equation 4. The measure compares the area of overlap between the prediction and true annotation to the combined area of both. A common threshold for positive detection of one instance is setting IoU to 0.5.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{4}$$

Confusion metrics can be calculated for desired class after setting the IoU threshold. An example of determining True positives (TP), False negatives (FN), and False positives (FP) based on IoU threshold is illustrated in Figure 8. TPs are true detections of the class of interest where the IoU threshold was met. FPs are detections that were made by the model, but without matching any ground truth annotation. FNs are class instances that were there, but could not be detected by the model.

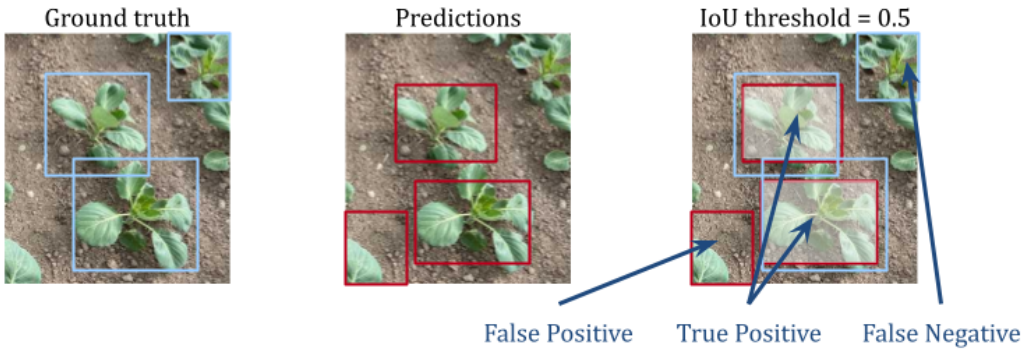


Figure 8: Object detection example of lettuce crop when setting IoU threshold to 0.5.

True negatives (TN) signifies all negative detections that were not made by the model. This metric is difficult to determine as there are many possible wrong detections. Therefore, TN is often skipped in object detection when evaluating performance on one class and is in this study likewise. It is worth mentioning that in other experimental settings for weed detection, it is common to define the weed class as the positives and plant class as the negatives, which enables finding a value for TN.

Model performance can be estimated from the confusion metrics. Accuracy is defined as the proportion of correct classifications of class instances (TP) among the total, calculated through Equation 5. Precision and recall are measures of relevance [9, 36]. Precision indicates successful detection rate and is defined as the proportion of correct detections (TP) among all detections of the class (TP and FP), Equation 6. Recall measures the effectiveness and is defined as the proportion of correct detections (TP) among all true objects (TP and FN), Equation 7). The F_1 score, Equation 8, is the harmonic mean of the precision and recall values [14].

$$Accuracy = \frac{TP}{TP + FP + FN} \tag{5}$$

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{6}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{8}$$

Calculating metrics based on one arbitrary IoU threshold can be disadvantageous as it does not capture the general performance of the trained model. In object detection, average precision (AP or mAP) has become a standard metric used in an increasing number of publications since the MS COCO dataset challenge was released. AP for one class is defined as the area below the precision versus recall curve, capturing a range of IoUs. The cure is created by calculating precision and recall at 11 different IoU thresholds, $\{0.0, 0.1, ... \ 0.9, 1.0\}$. The larger the area, the better the AP performance of a particular class. mAP signifies the AP averaged over all relevant classes.

$$\text{AP} = \int_0^1 P(R)\, dR \tag{9}$$

3 Related Work

Related work comparing CNN-based DL frameworks in vegetation tasks have been published in the literature and as open-source projects in recent years. Studies tend to adopt a similar experimental design. Following workflow has been proposed by Hasan *et al.* [11] after reviewing weed detection studies.

1. Data Aquisition (through data collection or online sources)
2. Data Preparation (including annotation, augmentation, synthesizing, coloring)
3. Image Pre-processing (e.g. denoising, enhancement, background reduction, fixing motion blur)
4. Training the CNN Model
5. Evaluation and Deployment of the model

Authors often report achieving high performances in weed detection when adopting this workflow. Some deep CNN-based models obtain accuracy or F1 score above 0.95, but either conduct experiments on small, tidy, and synthetic datasets, focus on plant detection instead of weed or use the image classification approach without localization. Studies that chose a similar approach to this thesis are described more in-depth in Sections 3.1 - 3.3 below.

3.1 Detecting weeds in cereal crops with Single Shot Detector

Dyrmann *et al.* [20] conducted a study intending to localize weeds among cereal crops where a high degree of occlusion was present in the visual imagery. The authors described a need to develop a solution for remote sensing of weeds based on computer vision in Danish weed management. Similar to Sweden, around half of the cultivated crops are cereals. The selected framework for this task was based on the SSD for which the authors argued could deal with the occlusion obstacle as well as identify different classes of weeds. For data acquisition, images were collected from seven fields growing maize, winter rye, and wheat, using a mobile camera as well as a high-resolution speed-stabilized camera mounted on top of an ATV. Over 1400 images were collected and prepared by annotating the weeds present using bounding boxes. In Figure 9, an example image from the Dyrmann *et al.* study is presented with annotations. Some weeds in their data are grass-like and show much resemblance to young cereals with long stringy leaves. Other weeds have broad leaves. Therefore, the authors separated the weeds into two classes; grass-like and broadleaf. Only around 3 % of the total weeds are classified as grass-like which creates an unbalanced classification problem.

Figure 9: Annotations of grass-like (red) and broadleaf (blue) weeds using bounding boxes presented in the Dyrmann *et al.* [20] study. The weeds are detected among cereal crops in Denmark.

The dataset was split into 1368 images with 13 177 bounding box annotations for training, and 51 images with 1119 annotations for testing the SSD model. More specifically, the SSD512 architecture using a VGG16 as backbone was used. The hyperparameters used for training were not presented.

In their results, Dyrmann *et al.* achieved a precision of 0.82, recall of 0.6, and F_1 score of 0.69 after setting the IoU threshold to 0.1 and predicting their independent test set. The authors explain that although the network can detect weeds correctly, despite that occlusion is high, many small weeds go undetected while larger weeds often get separate detections for their leaves. An example detection made by their model is shown in Figure 10, highlighting the difficulty with annotations, FP, and FN predictions. A more robust training set of images is needed to improve the performance which is not satisfactory according to the authors. SDD generally overrates the sizes of the objects and possibly a better suitable CNN framework could be implemented.

Figure 10: Prediction results made by Dyrmann *et al.* [20] achieving an F_1 score of 0.69. Ground truth annotations (blue and red) often overlap with the predictions (light blue), although IoU is low. Some misclassifications on background objects (FP) are present, and some objects go undetected (FN).

3.2 Recognition of weeds by training Mask R-CNN on real and synthetic data

In an attempt to build a weed detector that can recognize weed species among plants, Valicharla [36] implemented the Mask R-CNN framework in two different experiments; first using the small, public Crop/Weed Field Image Dataset (CWFID) originating from northern Europe, and secondly, on a larger synthetic dataset. One objective was to compare results from both experiments and whether an increase in acquired data through image augmentation can help mimic the variation of data that comes from the field. For both experiments, Valicharla set up the Mask R-CNN framework with ResNet50-FPN, pre-trained on the MS COCO dataset, as the baseline in the Detectron2 software platform.

In the first experiment, CWFID [37] dataset was used containing 60 RGB images collected from an organic carrot farm as well as masks for a 'plant' and a 'weed' class. Figure 11 shows an example image from this dataset together with its annotations. The image annotations were converted to the COCO annotation format where the 'iscrowd' parameter was used to define that there are several instances of one class in each image.

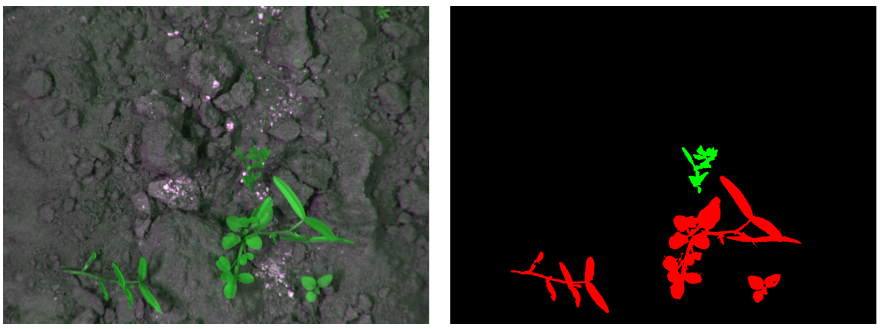


Figure 11: An image of carrot plants and weeds taken from the CWFID dataset with corresponding mask image to the right. Weeds are colored in red, and plants in green.

Before training, the dataset was split into 80 % training and 20 % test samples. Hyperparameters were set to learning rate 0.001, batch size 4, and epochs to 1000. The experiments were evaluated using mean AP metrics of both classes where Valicharla reported obtaining an AP of 0.80 when setting the confidence threshold to 0.9. Detections made by the model are illustrated in Figure 12. Although the AP achieved by the model is satisfactory (IoU is calculated from the overlapping bounding boxes instead of pixel-wise), the model fails to create a mask of the instances with high coverage as seen in Figure 12, to the left. Furthermore, there are two overlapping detections of plants that are unfavorable when calculating performance metrics and lowers the trust in the model. For Mask R-CNN, an improvement could be made by attempting to segment instances individually in the two classes instead of using crowd detection.

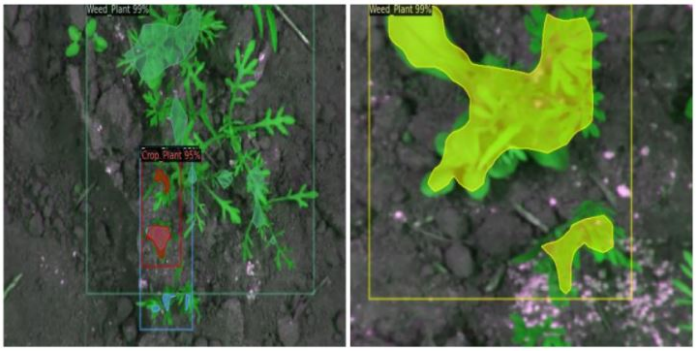


Figure 12: Detection result in two different images from the CWFID dataset produced by Mask R-CNN with ResNet50-FPN backbone in Detectron2. Image presented in the study by Valicharla [36].

The drawback using CWFID is that there are too few training samples for a deep CNN network, and the image settings are very specific with the gray background and enhanced green color in plants. The author also argues that for a robust weed detection model, more data has to be gathered or synthesized through manipulation and augmentation. Image augmentation is preferred by researchers as it avoids the data acquisition bottleneck which requires manual drawing of each instance in images. Therefore in a second experiment, Valicharla synthesized a dataset containing 1250 images where 80 types of weeds were cropped and pasted onto background images of grass. One example image is presented in Figure 13. At first, the 80 weeds were separated into separate classes but later merged to one super class. The Mask R-CNN framework was set up similarly to previous experiment, except from changing batch size to 2 and confidence threshold to 0.5. In the end, the obtained AP of weed detection in the synthetic dataset was 0.50, which is much less than previously.

In summary, Detectron2 is a flexible platform for setting up baseline frameworks for transfer learning, applicable on weed instance segmentation. The results from Valicharla study with Mask R-CNN did not produce robust results, especially for the synthetic dataset. More thought needs to be put into data acquisition, data preparation, training and evaluation.

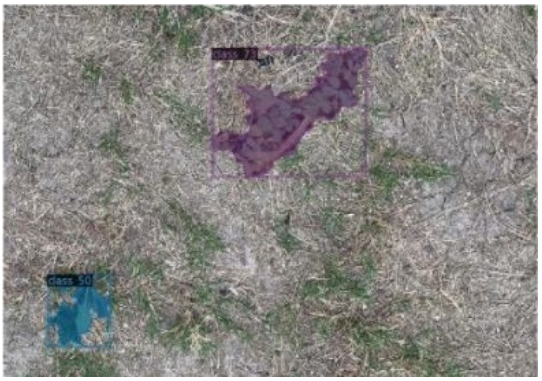


Figure 13: Image example from the synthetic dataset created by Valicharla [36]. Two weeds of class 50 and 73 are cropped on top of a background. The classes are later merged to one super class for weeds.

3.3 Localization of plants and weeds with Mask R-CNN

Aiming to improve localization of plants and weeds, as well as to separate different species, Champ *et al.* [8] conducted a study implementing instance segmentation. The objective of this study was to accurately predict the central point, or centroid, for a future weed control system based on mechanical or electric weeding. The hypothesis of the authors was that object detection frameworks that output bounding box detections, have poorer judgments of weeds that often form irregular shapes. Therefore, Champ *et al.* implemented the Mask R-CNN framework and compared centroid calculations from outputted polygon masks to bboxes, as well as estimated the removal rate that a potential weeding robot could achieve. The authors collected an image dataset comprising of 104 field images from French farmlands growing maize and bean crops with high-resolution Canon EOS cameras, mounted on an ATV. Manual labeling of polygons was performed using the COCO Annotator, outputting masks in COCO annotation format directly. Two separate classes of crops were labeled (long-leafed maize plants and broadleaf bean plants), as well as five separate classes of weeds. Four classes each consisted of a known species of weed that had been purposely cultivated, while the fifth weed class included the remaining weeds covering unknowns species. Figure 14 illustrates polygon masks that were drawn on a cultivated weed species with broad leaves. The classes had an uneven distribution of instances, ranging from 126 total instances of the maize class to 1370 total instances of the generalized weed class.

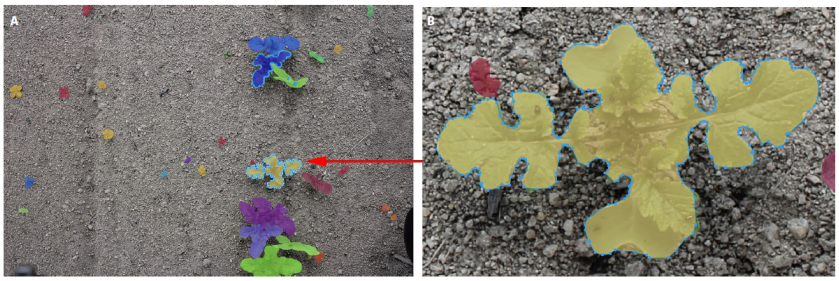


Figure 14: Polygon mask annotations of weeds and magnification of one broadleaf Black mustard instance made by Champ *et al.* [8].

The annotated dataset was split into 83 images with 2489 instances for training, and 21 images with 1018 instances for testing. Mask R-CNN with a ResNet50-FPN backbone was set up using the Detectron2 software platform. Hyperparameters were set to 40 epochs, batch size 2, and learning rate 0.001. The learning rate was increased from 0.0005 to 0.001 during a warmup period in the first epoch. In addition, batch normalization was implemented to reduce over-fitting. To further increase the training samples, images with their corresponding annotations were subjected to augmentation using random rotation, horizontal and vertical flips, as well as color, contrast, saturation, and brightness adjustments within the software. The performance of the model was evaluated primarily by calculating AP for each target class, and secondly by calculating the error in centroid estimation. In their results, the authors report achieving APs ranging from 0.15 to 0.85 for the different classes, see Table 1. The mean AP for all classes was 0.49. Furthermore, Champ *et al.* computed that the average centroid estimation error is 6.1 mm using the bounding box output versus 2.2 mm using the predicted mask center. An illustration of center estimation is presented in Figure 15.

Table 1: Mask R-CNN performances on different classes of plants and weeds obtained by Camp *et al.* [8].

Class	**Weed/Plant**	**Train instances**	**Test instances**	**AP**
Maize	Plant	98	28	0.85
Beans	Plant	405	49	0.59
Chamomille flower	Weed	362	333	0.27
White goosefoot	Weed	228	34	0.35
Black mustard	Weed	238	26	0.73
Ryegrass	Weed	290	46	0.15
Other	Weed	868	502	0.36

Figure 15: Centroid calculations produced from bounding box predictions (blue) versus polygon mask predictions (pink) illustrated by Champ *et al.* [8].

In addition, the authors find that the model is better at detecting larger instances of weeds and crops, while smaller ones often go undetected. This could mean that there is a bias towards the larger instances which needs to be adjusted for. Overall, this study shows the difficulty in accurate detection of different classes of vegetation in a real-world problem. The classes can easily divide in an unbalanced manner which influences the performance. Also, ryegrass which is part of the cereal crop family, here treated as a weed, seems to be poorly identified by Mask R-CNN.

4 Methods

The methods in this study attempt to follow the workflow suggested by Hasan *et al.* [11] in Section 3 while staying within project boundaries mentioned in Section 1.5. The experimental setup involved data preparation of acquired images and conversion of available instance masks to COCO annotation format, splitting the dataset to training, validation and test sets, setting up Mask R-CNN for training, and finally, evaluation using performance metrics.

4.1 Data Aquisition

A large variety of field images have been collected by Solvi originating from 15 farmlands located in Skåne (Southern) and Västra Götaland (Western Coast) county in Sweden. Images were taken using an integrated UAV with mounted high-resolution RGB cameras (DJI Mavic 1/2 Pro and DJI Phantom 4). The cameras are movement stabilized with Gimbal 3-axis tilt. Images were captured at 3 - 5 meters over the ground at varying surface and lightning conditions. A large image dataset of up to 10 000 RGB images has been generated in JPG as well as in TIFF format.

The TIFF images include geographical metadata and use real geospatial representation in coordinate reference system (CRS) 32633. The CRS for our dataset maps locations in UTM zone 33N, the area between 12 and 18 degrees east in the Northern Hemisphere (including southern lands of Sweden). Each pixel in the image is expressed as a shift in meters in relation to the origin of CRS 32633 (12 degrees east meeting the equator), and the images are tilted so that north is facing upwards. An example TIFF image is visualized in Figure 16.

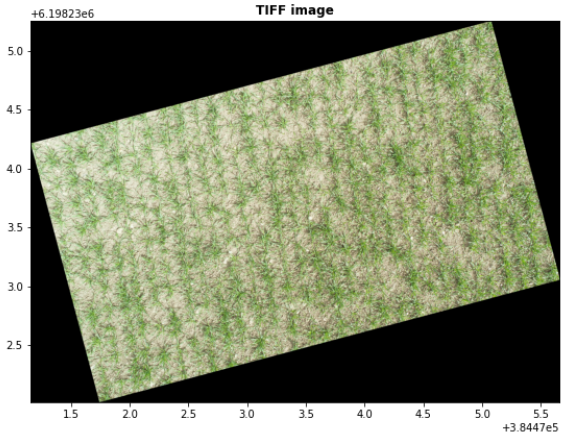


Figure 16: Raw TIFF image with geospatial representation (CRS 32633) in meters. The location of the image is 6 198 235 meters from the equator and 384 471 meters to the east of zone 33's meridian.

4.2 Data Preparation

In total, 70 images have been annotated by experts in precision agriculture at Solvi. Annotations were made by drawing polygons around instances of 'plant' and 'weed'

classes using the open-source geographic information system tool QGIS, and saved in GeoJson files. Polygon drawings were restricted to smaller regions within the high-resolution raw images, defined by bounding boxes in the GeoJsons. Examples of annotation regions in JPG and TIFF images are shown in Figure 17a and Figure 17b. No identification of different weed species was available.

For JPG images, the annotations are expressed in local pixel coordinates, while TIFF annotations are expressed in CRS 32633. The data preparation pipeline, therefore, had to be built to handle the two separate cases of input formats of images and GeoJsons. When parsing the dataset, JPG annotations have been read as-is, while TIFF annotations expressed in geographical coordinates have been projected by backward transform to local pixel space. Each point in the TIFF GeoJson was divided by the meter resolution and shifted back towards image origin (0,0).

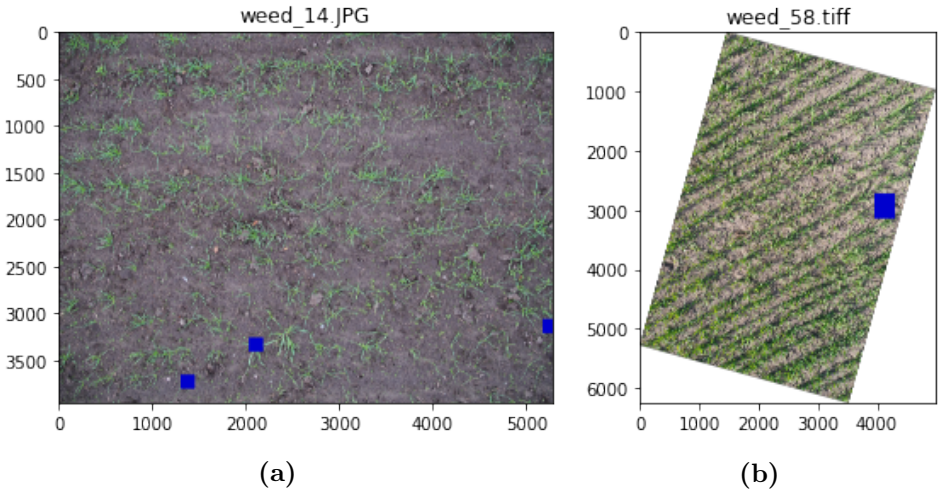


Figure 17: Bounding boxes within UAV images that contain annotations (colored in blue). The bounding boxes were cropped out and used as sample images to train and evaluate the Mask R-CNN model. (a) shows a raw JPG image and (b) a TIFF image converted from CRS 32633 to local pixel space.

A new dataset containing the annotated regions of images could be created by cropping raw images according to the bounding boxes defining them (visualized in Figure 17). Polygon mask entries were extracted for each sub-image by projecting polygons from the raw image space onto the space spanned by their corresponding bounding box. Example images from the cropped dataset are presented in Figure 18a - 18f where 'plant' and 'weed' instances are drawn out. In total, 200 annotated images were generated containing 3104 polygon instances which were balanced between the 'plant' and 'weed' classes. Image heights and widths vary, ranging from 150 to 670 pixels. The dataset had a large natural variation. This includes number of instances present, shapes and sizes of plants and weeds, color and texture of background surfaces, occlusion present, shades, as well as blurriness and image size.

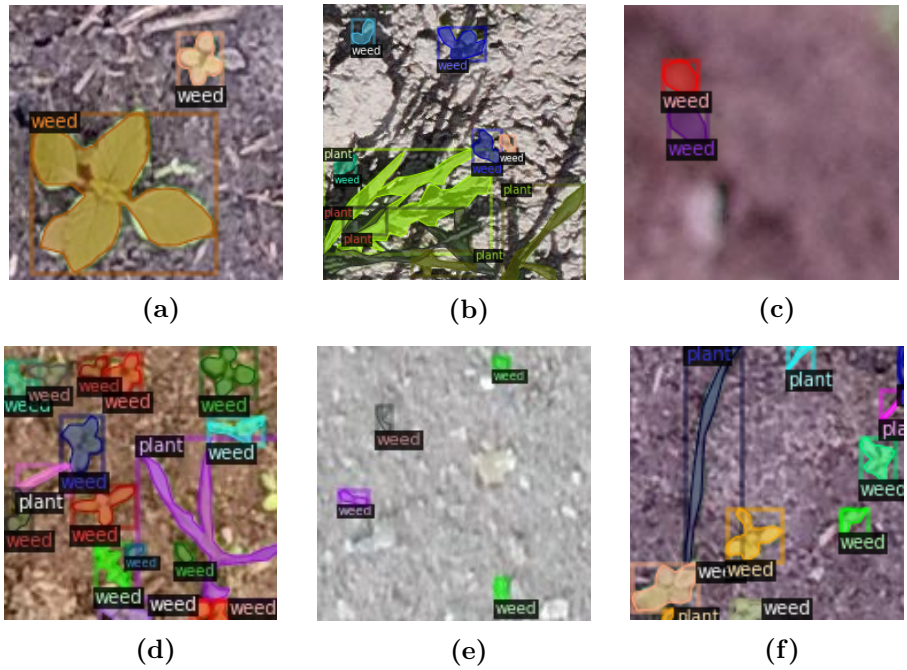


Figure 18: (a) - (f) Example subset of cropped UAV images with polygon annotations of 'weed' and 'plant' instances. There is a high variation in the the cropped image dataset is illustrated.

4.3 Training Mask R-CNN

The dataset was split into 70 % training (140 images with 2243 instances), 15 % validation (30 images with 425 instances), and 15 % test (30 images with 436 instances). Mask R-CNN was set up in three different experiments using ResNet50-FPN, ResNet101-FPN, and ResNeXt101-FPN backbones. The same hyperparameters were used for all experiments. The learning rate was set to 0.001, batch size to 2, max iterations to 10 000, warmup iterations to 500, and evaluation iterations to 500. The training was executed running on Nvidia GPU Tesla T4 with CUDA version 11.2. During training, the model performances were evaluated and tuned towards the validation set.

4.4 Evaluation

After training Mask R-CNN models with different backbones, performances were evaluated and compared in weed detection. Predictions were made on the independent test set that had 227 instances of weeds. Results with confidence above 0.5 were saved. Subsequently, confusion metrics, TP, FP, and FN, were obtained setting the IoU threshold to 0.5 from which accuracy, recall, precision, and F1 scores for the weed class could be calculated. The plant class has not been evaluated thoroughly in this work as it was not part of our main objective.

4.5 System Pipeline

The complete software workflow was implemented in Python and is illustrated in Figure 19. In the first data acquisition step, raw images in JPG and TIFF format, as well as GeoJson annotations, are read into the system. The subsequent data preparation phase consists of several modules which are described below. During this phase, the sub-image dataset is created from parsed raw images and annotations, and the output is converted to COCO annotation format, readable by Detectron2.

- `coordinates` - small functions for coordinate operations (e.g. projecting points from CRS to pixel space, calculating centroids of instances, or a bounding box surrounding a polygon).
- `visualization` - functions to read, visualize, and get image metadata mainly using PIL/Pillow library.
- `geojson` - contains a GeojsonParser class that reads GeoJson annotation files. Annotations in local pixel space are handled as-is, while annotations in CRS are transformed using the rasterio library that can process geographical raster data.
- `imagedb` - contains classes ImageDataset, BboxDataset, and PolygonDataset which handle and create new image datasets. A fourth class, DatasetDicts converts an image dataset to COCO format. ImageDataset parent class creates a dataset object from GeojsonParser output. BboxDataset inherits from ImageDataset and extracts bounding box annotations as well as crops raw images to sub-images. PolygonDataset inherits from ImageDataset and extracts polygon annotations as well as projects them to their corresponding bounding box by transforming the coordinates from the raw image pixel space to the sub-image. DatasetDicts extracts annotations from BboxDataset and PolygonDataset and saves them in COCO annotation format.

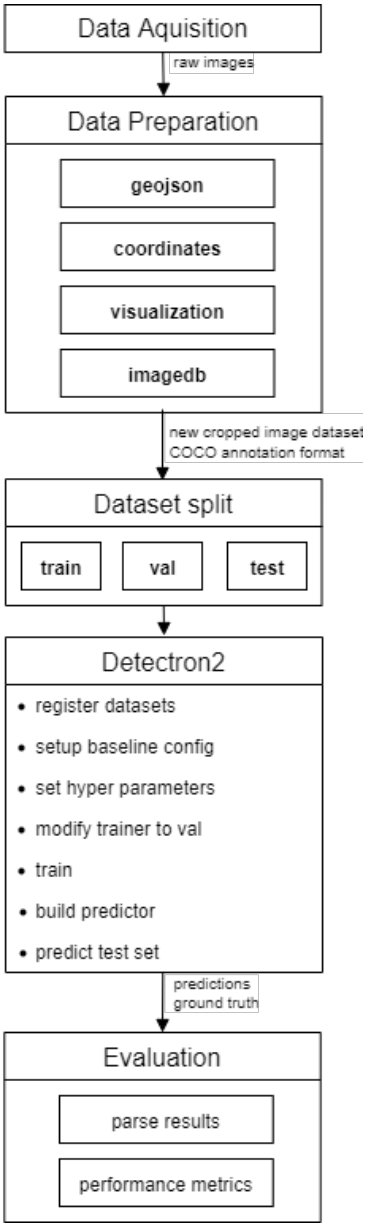


Figure 19: System pipeline when training and evaluating Mask R-CNN with the UAV dataset generated by Solvi.

The outputted DatasetDicts object contains the complete dataset. Before training a baseline model, DatasetDicts is split into three separate datasets for training, validation, and testing. Detectron2 software implemented in Pytorch library is set up to train a baseline model using DatasetDicts objects. The datasets are registered into the Detectron2 catalog and a baseline model (in our case Mask R-CNN baselines, pre-trained on the MS COCO dataset) is imported through a config file from the Detectron2 model zoo. Hyperparameters are tuned and the DefaultTrainer is modified to evaluate performance on the validation set after running each batch. A DefaultPredictor is built using the outputted model weights which are further applied to generate predictions of the test set images. The final step contains a small module for the evaluation process. First, a function reads the ground truth test set annotations together with the predictions made by Detectron2 (meeting desired confidence threshold) for one class of interest. A second function calculates the number of positive and negative detections by evaluating the IoU for a certain threshold. Precision, recall, accuracy, and F_1 score are calculated and outputted.

5 Results

Our results after training the three different Mask R-CNN frameworks to recognize weeds are presented in Table 2. Predictions above 0.5 confidence threshold were used and the IoU threshold for positive detection was set to 0.5. The best model according to our experiment used ResNet50-FPN as the backbone, followed by ResNet101-FPN and finally ResNeXt101-FPN. ResNet50-FPN obtained a precision of 0.83, recall of 0.61, F_1 score of 0.70, and accuracy of 0.54. The loss over the training iterations is presented in Figure 20.

Table 2: Model performances detecting weeds in UAV images evaluated on an independent test set selecting detection threshold 0.5 and IoU threshold 0.5.

Baseline model	TP	FP	FN	Precision	Recall	F1 score	Accuracy
ResNet50-FPN	138	28	88	0.83	0.61	0.70	0.54
ResNet101-FPN	127	27	100	0.82	0.56	0.67	0.50
ResNeXt101-FPN	123	26	104	0.83	0.54	0.65	0.49

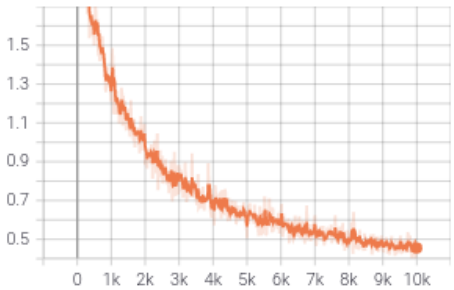


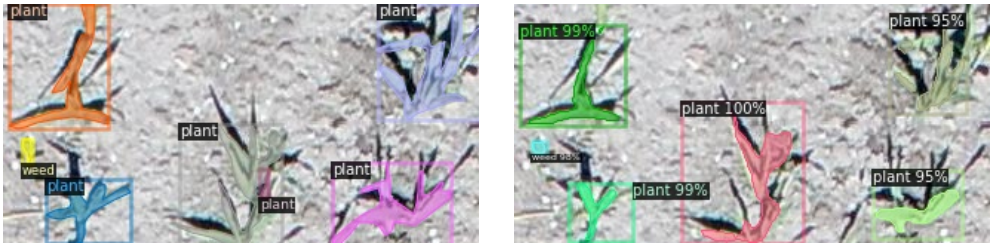
Figure 20: Loss over 10 000 iterations when training Mask R-CNN with Resnet50-FPN in Detectron2.

Overall, the performance differences are rather small across the three backbones. In the confusion metrics, mostly true positives are detected. Due to relatively few false positives, we obtain a satisfactory precision (0.82 - 0.83). False negatives, or missed detections, are higher. This leads to lower recall values (0.54 - 0.61) and a drop in overall F_1 score (0.65 - 0.70) as well as in accuracy (0.49 - 0.54).

Manually inspecting prediction results produced by the best model gives an understanding of the detection patterns. In Figure 21 - 24, predictions of plants and weeds compared to the ground truth annotations are illustrated for four different images taken from the independent test set. Figure 21 shows detections of weeds in a small-sized image. All weeds are identified although three unique weeds are segmented as one instance by the model. In Figure 22, one weed is present among six instances of plants, which the model detects. One small plant instance overlapping with a larger one is missed. In Figure 23 there are more weed and plant instances present in total. Here, one plant leaf and seven weed instances are not detected. Finally, in a large image shown in 24, there is an abundance of instances present, and likewise, an increased amount of mistakes made by the model. There is a rise in undetected weeds as well as plants, especially the ones of smaller size. In the top right corner, a large weed is classified as a plant. In contrast, a missing annotation of a plant located in the bottom right corner is correctly detected by the model.

(a) Ground truth **(b)** Predictions

Figure 21: Comparison between (a) ground truth and (b) ResNet50 predicted polygon masks in a small-sized test image containing weed instances. Three weeds are segmented as one larger weed instance by the model.

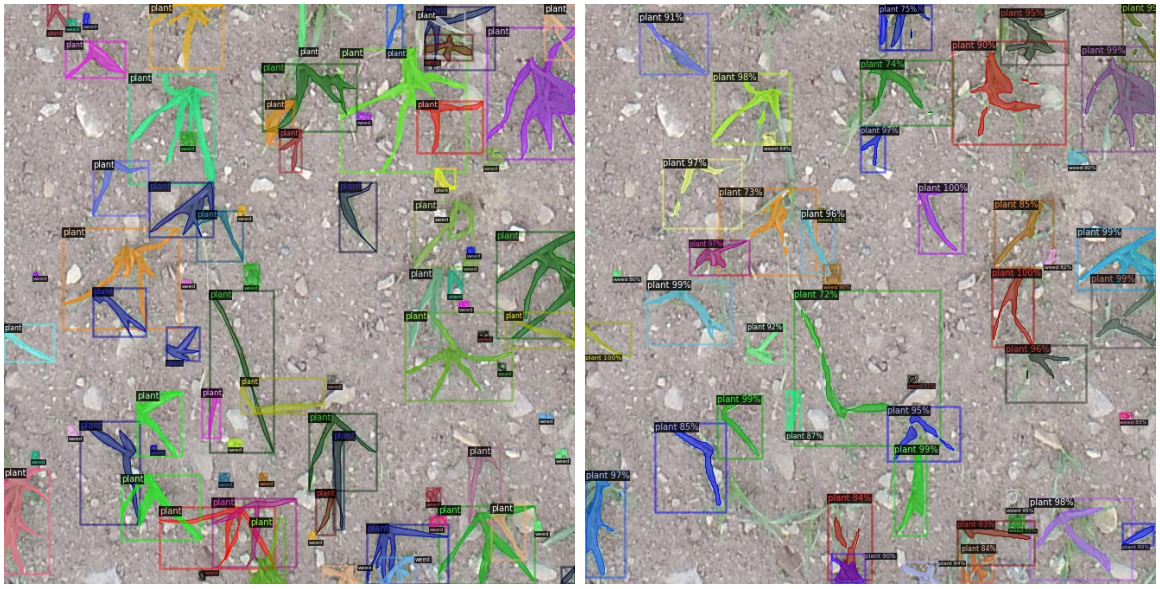


(a) Ground truth **(b)** Predictions

Figure 22: Comparison between (a) ground truth and (b) ResNet50 predicted polygon masks in a medium-sized test image with one weed instance among many plants. One tiny plant instance is not detected.

(a) Ground truth **(b)** Predictions

Figure 23: Comparison between (a) ground truth and (b) ResNet50 predicted polygon masks in a medium-sized test image with many weeds and plants present. Several weeds go undetected.



(a) Ground truth **(b)** Predictions

Figure 24: Comparison between (a) ground truth and (b) ResNet50 predicted polygon masks in a large test set image. Many instances of both classes are present. Here, many instances are not detected, especially the smaller weeds.

6 Discussion

In this work, we have trained a Mask R-CNN model to recognize weeds among cereal plants from UAV images. The best model that was obtained for the relatively small dataset used ResNet in 50 layers together with feature pyramid network (FPN) for feature extraction, see Table 2. Deploying ResNet-50 would lead to accurate detection of around 50 % of the weeds. Backbone networks such as ResNet-101 and ResNeXt-101, do not generate better performances on weed detection when using the same hyperparameters. This suggests that for our type of image complexity, adding depth to the network does not improve the results.

Compared to related work in the literature (some of which were presented in Section 3, in reference studies [3, 9, 16, 17] and review articles [2, 7, 10, 11]), we obtain similar performances while at the same time encountering the same challenges. In most cases, the instances of weeds are correctly classified and obtain a mask that corresponds to the object boundaries. Yet, some prevalent limitations are correct segmentation of instances that experience occlusion (as demonstrated in Figure 21 and Figure 22), difficulty to detect weed instances due to them being small in size (as demonstrated in Figure 23 and Figure 24), or due to too many instances being present overall in the image (demonstrated in Figure 24). The algorithm could be inadequate to produce accurate anchors for objects when too many instances are present in large images, or when there is too much variation in the instance sizes. Furthermore, a common source of uncertainty might come from the subjectivity during the annotation procedure which further complicates the understanding of occlusion [29]. In Figure 21a, three weed instances are described using separate masks, while in Figure 23a, one annotation of a very large weed instance might have been created covering more than one unique instance. The opposite can be seen in Figure 22a where a small plant within a larger plant instance could potentially be annotated as one instead of two. Including images with higher blurriness or low image resolution in the training and testing datasets (such as the image shown in Figure 18c) may further limit our performance.

On the whole, what stands out from our study is that we have been able to introduce a variety of conditions within the small dataset that we have used. Although contributing to lower accuracy, our model is more robust in handling different experimental settings and therefore better reflects the real-world problem.

7 Conclusion and Future Work

Weed control which is less dependent on the use of herbicides is a requirement for sustainable agriculture. In Sweden, Jordbruksverket (The Swedish Board of Agriculture) is pushing organizations and companies to design new data-driven solutions. A weeding management system that can be executed among the most commonly cultivated crops in the Nordics - cereals, would make a large impact on pesticide reduction in Sweden. The system can look as follows. Imagery data at any desired stage in cultivation is gathered using a UAV and weed patches are identified from the images by applying the weed detector model. Post-processing using geographical metadata is added to the pipeline to construct a weed growth blueprint of the complete field. The blueprint can thereafter serve as a steering file for the automated sprayer vehicle which passes the field in close proximity.

Together with Solvi, we aimed to fill a gap in available agriculture technology for designing such a system. We have built a remote sensing model targeting weeds growing among cereals and achieved a precision of 0.83, recall of 0.61, F_1 score of 0.70, and accuracy of 0.54. Our final Mask R-CNN model with ResNet-50 backbone demonstrates that it has the capability of recognizing unique instances of weeds and plants in diverse conditions, but is yet not mature enough to be deployed at a large scale. Using the model would serve us in identifying around 50 % of the instances. In all

likelihood, the model would be biased towards the detection of the larger individuals. In our case that might be solved when passing the field the forthcoming round, phasing out weeds as they grow. As a start, the coupled sprayer could be adjusted to assume more weeds are present than it detects and uses slightly higher volumes of pesticides to compensate for the low recall rate. Even though more volumes would be spent than what we initially aimed for, the solution is favorable compared to covering whole fields with larger amounts of chemicals. In future work, an enhanced version of the weed detector could be built with several improvements in the data collection process as well as in the system pipeline to further minimize the pesticide need.

7.1 Improving input data quality with image pre-processing

Some image pre-processing techniques can be applied to enhance the features of raw images and improve image quality before training. Primarily, limiting the sub-images to smaller croppings, e.g. by further splitting the larger images, could correct for the number of false-negative detections. Further on, detection could be facilitated by first applying contrast enhancement, color space transformation, normalization, or background reduction [2]. Finally, denoising techniques can be applied to improve image quality, especially if there is a limitation in collecting ultra-resolved images with more advanced camera and UAV hardware.

7.2 Increasing sample dataset

Vast amounts of data are typically required to properly train deep computer vision frameworks and reduce over-fitting [7]. Although more input data could be generated by annotating the available images from the large dataset, this would be a very time-consuming procedure. There are several methods to incorporate more sample images that bypass manual labeling. Common techniques that do not require a substantial effort to add to the pipeline compose of creating synthetic images and/or image augmentation. First, croppings of known weeds and plants can be pasted on top of a variety of backgrounds [36]. Next, augmentation can be performed through operations such as flipping and resizing, adjustment of contrast, saturation, and brightness, or introducing other types of image distortions. In this way, the reference image dataset is inflated while at the same time keeping already made masks [7, 16]. Ultimately even more advanced methods to automatize the annotation procedure exist, where specialized annotation AI assistants can learn the basics of mask drawings after seeing a few manual annotation examples [17].

7.3 Revising the CNN framework

The choice regarding computer vision framework and backbone CNN architectures within this project have been limited to Mask R-CNN and ResNet configurations as they are implemented and available in Detectron2. Nevertheless, Mask R-CNN with ResNet backbone is a popular choice for machine vision tasks. Faster R-CNN and Mask R-CNN are known for achieving higher accuracy compared to YOLO and SDD where a shorter inference speed is valued higher. To extend this experiment to test

other backbone configurations, the Python platform would need to be switched (e.g. to using Pytorch from scratch) - a procedure where the gain might not exceed the effort. Instead, other baseline models available in the Detectron2 model zoo could potentially be trialed. For example, the Mask R-CNN pre-trained on the COCO dataset together with large-scale jitter and longer training schedule, Mask R-CNN pre-trained on the LVIS dataset [38], or the object detection framework named Cascade R-CNN [39]. Improvements to the training procedure could additionally be attempted. For example, increasing the training iterations while at the same time incorporating early stopping could help reduce over-fitting if a larger annotated dataset would be available. To improve correct segmentation, enabling crowd detection by setting the 'iscrowd' parameter in Detectron2 could be useful. Further on, separate models could be trained on blurry images versus high resolved ones to optimize results for the separate scenarios. Finally, implementing calculations of AP could be better suitable for performance estimation.

In the end, we believe the solution to machine vision tasks lie in incorporating big data (data with high variety, veracity, volume, velocity and value) rather than optimizing the performance on one explicit and unique task, and should therefore be the future focus when developing a robust and transferable weed detector.

References

[1] Food FAO et al. The future of food and agriculture - Trends and challenges. *Annual Report*, 296, 2017.

[2] Aichen Wang, Wen Zhang, and Xinhua Wei. A review on weed detection using ground-based machine vision and image processing techniques. *Computers and electronics in agriculture*, 158:226–240, 2019.

[3] Bo Liu and Ryan Bruch. Weed detection for selective spraying: a review. *Current Robotics Reports*, 1(1):19–26, 2020.

[4] Åke Fridolfsson and Thomas Börjesson. Application for funding - site specific weed control. *Jordbruksverket*, 2019.

[5] Peter Einarsson and Anki Bergström. Kemiska bekämpningsmedel i jordbruket – fakta om användningen i Sverige 1981-2016. *Naturskyddsföreningen*, 2017.

[6] Francisca López-Granados. Weed detection for site-specific weed management: mapping and real-time approaches. *Weed Research*, 51(1):1–11, 2011.

[7] Teja Kattenborn, Jens Leitloff, Felix Schiefer, and Stefan Hinz. Review on convolutional neural networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:24–49, 2021.

[8] Julien Champ, Adan Mora-Fallas, Hervé Goëau, Erick Mata-Montero, Pierre Bonnet, and Alexis Joly. Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots. *Applications in plant sciences*, 8(7):e11373, 2020.

[9] S. Umamaheswari, R. Arjun, and D. Meganathan. Weed detection in farm crops using parallel image processing. In *2018 Conference on Information and Communication Technology (CICT)*, pages 1–4. IEEE, 2018.

[10] Efthimia Mavridou, Eleni Vrochidou, George A. Papakostas, Theodore Pachidis, and Vassilis G. Kaburlasos. Machine vision systems in precision agriculture for crop farming. *Journal of Imaging*, 5(12):89, 2019.

[11] Mahmudul A. Hasan, Ferdous Sohel, Dean Diepeveen, Hamid Laga, and Michael GK. Jones. A survey of deep learning techniques for weed detection from images. *Computers and Electronics in Agriculture*, 184:106067, 2021.

[12] Dian M Bah, Adel Hafiane, and Raphael Canals. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote sensing*, 10(11):1690, 2018.

[13] Solvi - all-in-one solution for drone-based crop monitoring. `https://solvi.ag/features`. Accessed: 2021-07-04.

[14] Jialin Yu, Shaun M Sharpe, Arnold W Schumann, and Nathan S Boyd. Deep learning for image-based weed detection in turfgrass. *European journal of agronomy*, 104:78–84, 2019.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[16] Xiaoyang Liu, Dean Zhao, Weikuan Jia, Wei Ji, Chengzhi Ruan, and Yueping Sun. Cucumber fruits detection in greenhouses based on instance segmentation. *IEEE Access*, 7:139635–139642, 2019.

[17] Shuangyu Xie, Chengsong Hu, Muthukumar Bagavathiannan, and Dezhen Song. Toward robotic weed control: Detection of nutsedge weed in bermudagrass turf using inaccurate and insufficient training data. *arXiv preprint arXiv:2106.08897*, 2021.

[18] The Swedish Plant Protection Council. *Jordbruksverket*. `https://jordbruksverket.se/languages/english/the-swedish-plant-protection-council`, 2021.

[19] Agroväst - sustainability and profitability for green industries. `https://agrovast.se/`. Accessed: 2021-03-12.

[20] Mads Dyrmann, Søren Skovsen, Morten Stigaard Laursen, and Rasmus Nyholm Jørgensen. Using a fully convolutional neural network for detecting locations of weeds in images from cereal fields. *The 14th International Conference on Precision Agriculture*, pages 1–7, 2018.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. *Springer*, pages 740–755, 2014.

[22] Yuxin Wu et al. Detectron2. `https://github.com/facebookresearch/detectron2`, Latest release: 2020-11-06.

[23] Brian D. Ripley. Pattern recognition and neural networks. *Cambridge University Press*, 2007.

[24] Mahamad Nabab Alam. Codes in MATLAB for training artificial neural network using particle swarm optimization. *Research Gate*, pages 1–16, 2016.

[25] Van Hiep Phung and Eun Joo Rhee. A deep learning approach for classification of cloud image patches on small datasets. *Journal of information and communication convergence engineering*, 16(3):173–178, 2018.

[26] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[27] Ross Girshick. Fast R-CNN. *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[29] Kavir Osorio, Andrés Puerto, Cesar Pedraza, David Jamaica, and Leonardo Rodríguez. A deep learning approach for weed detection in lettuce crops using multispectral images. *AgriEngineering*, 2(3):471–488, 2020.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. *European conference on computer vision*, pages 21–37, 2016.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[34] Osmar Luiz Ferreira de Carvalho, Osmar Abílio de Carvalho Júnior, Anesmar Olino de Albuquerque, Pablo Pozzobon de Bem, Cristiano Rosa Silva, Pedro Henrique Guimarães Ferreira, Rebeca dos Santos de Moura, Roberto Arnaldo Trancoso Gomes, Renato Fontes Guimarães, and Díbio Leandro Borges. Instance segmentation for large, multi-channel remote sensing imagery using Mask-RCNN and a mosaicking approach. *Remote Sensing, Multidisciplinary Digital Publishing Institute*, 13(1):39, 2021.

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[36] Sruthi Keerthi Valicharla. Weed recognition in agriculture: A Mask R-CNN approach. *The Research Repository at West Virginia University*, 2021.

[37] Sebastian Haug and Jörn Ostermann. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. *European conference on computer vision*, pages 105–116, 2014.

[38] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019.

[39] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.