Beyond mAP: Towards practical object detection for weed spraying in precision agriculture

Adrian Salazar-Gomez1,3 Madeleine Darbyshire2,3 Junfeng Gao1,3 Elizabeth I Sklar1,3 Simon Parsons2,3

Abstract—The evolution of smaller and more powerful GPUs over the last 2 decades has vastly increased the opportunity to apply robust deep learning-based machine vision approaches to real-time use cases in practical environments. One exciting application domain for such technologies is precision agriculture, where the ability to integrate on-board machine vision with data-driven actuation means that farmers can make decisions about crop care and harvesting at the level of the individual plant rather than the whole field. This makes sense both economically and environmentally. This paper assesses the feasibility of precision spraying weeds via a comprehensive evaluation of weed detection accuracy and speed using two separate datasets, two types of GPUs, and several state-of-the-art object detection algorithms. A simplified model of precision spraying is used to determine whether the weed detection accuracy achieved could result in a sufficiently high weed hit rate combined with a significant reduction in herbicide usage. The paper introduces two metrics to capture these aspects of the real-world deployment of precision weeding and demonstrates their utility through experimental results.

Index Terms—precision agriculture, automated weeding, computer vision, object detection

I. INTRODUCTION

The current agricultural approach to weeding in arable crops is to spray an entire field with a selective herbicide that kills the weeds, but does not harm the crops. Such a broadcast spraying approach is easy to deliver—requiring only a sprayer to dispense the herbicide—but wasteful, since much of the area sprayed does not contain weeds, being either bare earth or crops (see Figure 3). Precision agriculture aims to use ideas from artificial intelligence (AI) and robotics to create agricultural solutions that can be delivered at the level of individual plants, instead of entire fields. An important application within precision agriculture is automated weeding, which aims to detect and target individual weeds by precisely delivering herbicide [1] on the weeds (avoiding wastage) or by eliminating weeds using a laser [2] or mechanical tool [1]. In the work presented here, we are concerned with precise delivery of herbicide in a real-world farm setting.

![Fig. 1: A top-down view of an example of a small precision sprayer. The direction of travel is indicated in the drawing. Each camera captures its own image stream over time. Each row of images represents the set captured by the cameras at the same point in time; each column represents the passage of time as the tractor drives. The images here do not overlap.](image-url)

Now, it is clear that any approach to automate weeding needs to identify the weeds, and there have been numerous attempts to use computer vision to do this (see Section II), many using AI methods based on machine learning (ML). However, most of this work has treated automated weeding as purely a computer vision problem: datasets are collected and annotated, object identifiers are trained, and the resulting models are optimised with respect to accuracy and/or mean average precision (mAP) for classifying crops and weeds. We argue that while these metrics are important, there are additional measures that can contribute to assessing the feasibility of ML models for use in precision spraying. Here we describe three of them.

First, we introduce weed coverage rate (WCR), which identifies the proportion of the weeds that could be targeted by a sprayer which is activated using the model. WCR depends on the accuracy of the model, but also takes into account the resolution of a spray. Second, we define the area sprayed—the area covered in herbicide by the precision sprayer—which helps determine the savings in herbicide compared with current practice. Third, we measure inference speed to ascertain whether the approach is practical—because you can’t just pile more GPUs on-board a tractor operating in an open field, where issues like compute power and energy consumption come into play. Real-world automated weeding (see Figure 1) will need to process many images very rapidly, so here we use inference speed as a proxy for comparing the practicality of different ML approaches.

The primary contribution of this paper is to assess the feasibility of precision spraying by comparing a selection of...
standard ML-based vision methods applied to weed detection using not only mAP and inference speed but also the new metrics we have introduced: WCR and area sprayed. Section II briefly describes the state-of-the-art in ML-based vision approaches applied to agriculture. Section III explains our methodology, and Section IV details the experiments we conducted which form the basis of our comparison. Section V analyses these results, and Section VI closes with conclusions.

II. RELATED WORK

Initial approaches to weed detection used machine learning algorithms with handcrafted features based on differences in colour, shape or texture. [3] extracted local binary patterns with support vector machines for plant discrimination. This method generally requires a relatively small dataset for model development, but might fail to generalise under different field conditions. Deep learning-based methods for computer vision are increasingly popular because they offer an end-to-end weed detection solution with better generalisation. Modern object detectors such as YOLOv3 and CenterNet have been used to detect weeds in sugar beet fields and evaluated using mAP as the main metric [4] [5]. Other deep learning-based approaches locate weeds using segmentation where the evaluation was based on the number of pixels in an image classified correctly [6] [7].

Following on from work on weed detection, robotic weed control systems, that enable precision weed management, can be studied. ‘Spot’ or ‘selective’ spraying, the application case in this paper, involves switching an individual nozzle ON or OFF to deliver chemicals only to the weeds [8] [9]. However, most studies focus solely on detection without exploring the next step: how those detections are used to target an actuator to eradicate the weeds. Moreover, few studies attempt to measure performance in terms of real-world outcomes such as weeds sprayed and total herbicide used. Consequently, there is a lack of understanding around what level of detection accuracy is required to achieve adequate real-world performance. Other than spraying, robotic mechanical precision weeding [1] is an alternative, suitable for weed management in organic farming, but typically sacrificing operational efficiency compared with spraying. Our study estimates the most suitable deep learning-based object detector by considering multiple datasets from different locations. Our assessment of detectors is measured by balancing detection accuracy with inference speed, weed coverage rate and area sprayed for a robotic weed sprayer in a field environment.

III. METHODOLOGY

The overall scenario we consider is shown in Figure 1. A boom, mounted on the back of a farm vehicle, carries both cameras and nozzles for dispensing herbicide. The cameras are fixed a short distance ahead of the spray nozzles. This allows time for the weeds to be detected in an image before the corresponding area passes beneath the nozzles. These detections can then be used to determine whether a particular nozzle needs to be switched on or off at a particular time.

Object detection algorithms locate items of interest in images and demarcate them using bounding boxes. Deep learning-based object detectors learn to find objects using a training set of images accompanied by ground-truth labels. After this process, the trained detector is evaluated on a test set of labelled images. A conventional metric to evaluate the performance of object detectors is mean Average Precision (mAP). To calculate mAP, first, a threshold is defined for the Intersection over Union (IoU) value and this is used to distinguish between true positive (TP), false positive (FP) and false negative (FN) detections. IoU measures the accuracy of the predicted bounding box circumscribing the object identified and is equal to $(G \cap D) / (G \cup D)$, where $G$ is the area of the ground-truth (labelled) bounding box and $D$ is the area of the bounding box predicted by the model.

Precision (P) is the proportion of correctly identified objects over all objects identified, $TP / (TP + FP)$. Recall (R) is the proportion of correctly identified objects over all identified objects (correct or incorrect), $TP / (TP + FN)$. These metrics are used to compute average precision (AP):

$$AP = \sum_{n} (R_n - R_{n-1}) P_n$$

where $n$ is the IoU threshold rank. Because AP is dependent on the IoU threshold value, mean AP or mAP takes into account different threshold values and averages the AP for each of them. In this article, we follow the COCO standard [10] and calculate mAP using 10 IoU thresholds with a 0.50 : 0.95 range.

While mAP is a good measure of the performance of an object detector, it focuses on identifying the object very precisely. In practice, for a sprayer, the level of precision for spraying is limited by the spray nozzle, and, as we will see, it is possible for an object detector that has a relatively low mAP to still be good enough to be effective in ensuring that weeds are covered with herbicide.

To get a better idea of how the precision of the detector impacts the precision of spraying, we have devised a new metric which we call weed coverage rate (WCR). This estimates how many of the weeds in the test data would be sprayed with a particular nozzle setup in the sprayer, rather than how many are detected. We start by modelling a simplified spraying scenario using images of a field. Taking the $i$-th image in a dataset with $I$ images, we imagine the sprayer is moving parallel to the width of that image ($W_i$). The spray boom with nozzles arranged along its length is then parallel to the height of the image ($H_i$) as illustrated in Figure 2. The precision of spray nozzle can be varied to spray from 500mm wide, as in broadcast spraying, to around 100mm for precision spraying. In our model, we vary the...
number of nozzles \((n)\) from one to four\(^1\) by splitting the image along its height into \(n\) strips, each with width \(W_i\) pixels and height \(H_{ij}\) pixels, where a strip represents the area that each of the nozzles could spray.

The \(i\)-th image in the dataset has \(J_i\) detected weed bounding boxes, each with width \(w_{ij}\) and height \(h_{ij}\). For each strip that intersects with a detected weed bounding box, a spray area \(B_{i,k}\) is proposed in that strip, where \(B_{i,k}\) is the \(k\)-th spray area proposed in the \(i\)-th image. \(B_{i,k}\) has the height of the strip, and its width is \(max(H_{ij}/n, w_{ij})\)\(^2\). In the \(i\)-th image, there are \(K_i\) of these spray areas proposed. To derive \(S_i\), which is total spray area within the \(i\)-th image, to the exclusion of overlapping regions, the union of the proposed spray areas is calculated:

\[
S_i = \bigcup_k B_{i,k}.
\]

where both \(S_i\) and \(B_{i,k}\) are measured as the number of pixels enclosed in the area. The generation of \(S_i\) is illustrated in Figure 2.

The spray area is larger than just the bounding boxes because the size is partly determined by the width of the spray (compare Figures 2a and 2b). WCR differs from mAP because that additional area, while increasing the herbicide used, and decreasing mAP, will sometimes “hit” weeds that have evaded detection. WCR is defined as follows. A weed is counted as having been sprayed if it is wholly contained in the spray area:

\[
Sprayed(G_{i,m}) = \begin{cases} 
1, & \text{if } G_{i,m} \subseteq S_i, \\
0, & \text{otherwise}
\end{cases}
\]

where \(G_{i,m}\) is the area of the \(m\)-th ground truth weed bounding box in image \(i\). Then the WCR for a dataset is the fraction of the weeds that are counted as having been sprayed:

\[
WeedCoverageRate(I) = \frac{\sum_{i \in I} \sum_{m \in M_i} Sprayed(G_{i,m})}{\sum_{i \in I} M_i} \times 100, \tag{3}
\]

where \(M_i\) is the total number of ground truth weed detections in the \(i\)-th image. In addition, we compare the area that has been sprayed with the total area of all the images in the dataset:

\[
AreaSprayed(I) = \frac{\sum_i S_i}{\sum_i W_i \times H_i} \times 100. \tag{4}
\]

In order to establish the requirements in terms of frame-rate, we need to make some assumptions about the design of a sprayer. The boom on a typical sprayer is 24m, and the recommended height to operate the boom above the crop canopy is 500mm \([11]\). We established empirically that, at this height, a typical camera with a 1.8 : 1 aspect ratio (see below), can cover 670mm \(\times\) 380mm. The maximum speed of a sprayer is widely taken to be around 15\(mph\), or 6.7\(m/s\) (see, for example, \([12]\)) to prevent spray drift. With the long edge of the image aligned along the spray boom, we need 35 cameras, and each camera can cover 380\(mm\) in the direction of travel. At 6.7\(m/s\), we will need to process 28 frames per second (assuming no overlap) so, across all 35 cameras, the required frame-rate will be 980. With the short edge of the

\(^1\)In a picture taken with a standard deployment camera, such as a Realsense D435i with a focal length of 1.93mm and an F-number of f/2.0, and a standard distance to the crop canopy (~500mm), the longer side of the pictures will cover ~670mm and the shorter will ~380mm. Depending on the configuration of the spraying system, such as the pump pressure, the nozzle would spray at different precisions.

\(^2\)This is because most of the sprayers have a circular sprayer pattern, and the minimum spraying area will be a square where the sides have the dimension of the height.
image aligned along the boom we would need 64 cameras and process 16 frames a second per camera, giving a required frame-rate of 1024 FPS.

IV. EXPERIMENTS

A. Datasets

We used two weed/crop datasets: the Lincoln beet (LB) dataset\(^3\) which we collected and annotated as part of this work and the Belgium beet (BB) dataset from [4]. Both datasets contain pictures from fields in which sugar beet was grown commercially and the images contain pictures of beets and weeds with their respective bounding box locations. The BB dataset consists of 2506 images of 1800 × 1200 pixels. The LB dataset consists of 4405 images of 1920 × 1080 pixels. The LB images were extracted from videos\(^4\) recorded at different points in time, in three different sugar beet fields. The data collection spanned May–June 2021, where each field was scanned, at minimum, on four different dates a week apart to record weeds at different stages of growth. For all the data collection sessions, the distance from the camera to the ground was approximately 500mm. Two cameras were used: one with 12 megapixels, 26mm focal length and an f1.6 aperture; the other with 64 megapixels, 29mm focal length and an aperture of f2.0. The original size of pictures from both cameras was 2160 × 3840 pixels. The fields used for the LB dataset are near Lincoln, UK, at different locations with varying conditions as to the type of soil, distribution of the plants, and weed varieties. There are no previous results using the LB dataset as this paper introduces it. In [4], an overall mAP of 0.829 was achieved using the BB dataset. This was composed of a mAP of 0.761 for the weed class and 0.897 for the sugar beet class. Fig. 3 shows examples of the BB dataset and three fields used to create the LB dataset.

Both datasets present different item distributions and visibilities. The BB dataset has a lower number of items per picture than the LB dataset. In terms of visibility, the items in the BB dataset are near Lincoln, UK, at different locations with varying conditions as to the type of soil, distribution of the plants, and weed varieties. There are no previous results using the LB dataset as this paper introduces it. In [4], an overall mAP of 0.829 was achieved using the BB dataset. This was composed of a mAP of 0.761 for the weed class and 0.897 for the sugar beet class. Fig. 3 shows examples of the BB dataset and three fields used to create the LB dataset.

Both datasets present different item distributions and visibilities. The BB dataset has a lower number of items per picture than the LB dataset. In terms of visibility, the items in the BB dataset are near Lincoln, UK, at different locations with varying conditions as to the type of soil, distribution of the plants, and weed varieties. There are no previous results using the LB dataset as this paper introduces it. In [4], an overall mAP of 0.829 was achieved using the BB dataset. This was composed of a mAP of 0.761 for the weed class and 0.897 for the sugar beet class. Fig. 3 shows examples of the BB dataset and three fields used to create the LB dataset.

3The dataset is available at the following link: https://github.com/LAR/lincolnbeet_dataset
4The video frames converted to images were separated by enough frames that we could avoid repeated images in the dataset.

TABLE I: Characteristics of the BB and LB datasets at dataset level (top) and at item level (bottom)

|                    | BB dataset | LB dataset |
|--------------------|------------|------------|
| Number of images   | 2506       | 4402       |
| Number of items    | 5578       | 3924       |
| Average items per picture | 2.22 | 8.915 |
| Average percentage of the bounding box that is occluded | 0.02187 | 0.0176 |
| Average area of the image occupied by bounding boxes | 0.09883 | 0.0717 |

B. Data preparation

For the experiments, each dataset was randomly split into training, test and validation sets with 70%, 20%, and 10% of the dataset images, respectively. For training and testing, we resized the images and labels in both datasets to make the large side of the image fit in a 640-pixel dimension while maintaining the width/height ratio of the raw images.

C. Models and Parameters

For the identification of weeds, we implemented one-stage detectors and two-stage detectors. Two-stage detectors tend to produce more cautious predictions than the one-stage detectors. However, one-stage detectors tend to be faster. We implemented both approaches to determine whether the speed of the one-stage detectors or the cautious predictions of the two-stage detectors would have any impact on the metrics we use [13]. The one-stage models are Yolov5m [14], Yolov3 [15] and Yolov4 [16], where all three models use Darknet-53 [15] (DN-53) as a backbone. The two-stage detectors are based on Faster R-CNN [17] with three different backbones: one with a ResNet-50 backbone [18] and a Feature Pyramid Network [19] (FPN) neck; namely, ResNet-50-FPN (R-50-FPN), one with a ResNet-101-FPN [18] (ResNet-101-FPN), and another detector with a ResNeXt-101-FPN [20] (Rx-101-FPN). During training, for the one-stage and two-stage models, the optimiser was stochastic gradient descent (SGD), the learning rate was \(1 \times 10^{-2}\), the momentum was 0.937, and the learning decay was \(5 \times 10^{-4}\).
In both model varieties, the networks were pre-trained on the COCO dataset [10]. For testing, the confidence threshold was 0.05, and the intersection over the union threshold was 0.5. These values are obtained from a grid search maximising the mAP. For the sake of consistency, no data augmentation was applied in the training of these models, although, using data augmentation may improve accuracy.

The models were trained on a GTX2080Ti processor with 11GB of VRAM. Two GPUs with the same software configuration were used to evaluate the inference time: the GTX2080Ti and a Tesla T4 with 16GB of VRAM. The Tesla T4 has a ruggedized case and integrated battery which makes it more suitable for deployment in the field than the GTX2080Ti, which is not ruggedized or battery powered.

For all detectors, the batch size (the number of images fed to the models simultaneously) used for training was the maximum number of images that can fit in the GPU’s VRAM, alongside the detector. The number of epochs for training was 300. For each model, the model weights used for testing were the ones with the highest mAP on the validation set during the training process.

D. Results

We evaluated the trained models in several ways across both the BB and LB datasets and report both mAP as well as our new metrics: inference speed in frames per second (FPS), weed coverage rate (WCR) and area sprayed. Analysis of results and discussion are given in the next section (V).

Table II provides a conventional evaluation giving mAP and speed of inference with batch size = 1. The approaches with the best performance on these metrics are highlighted. Table III also provides inference speed, but exploits the ability of the models and the hardware they are run on to operate in parallel. Results are given for a range of batch sizes. Table IV reports weed coverage rate and area sprayed for one to four nozzles. Figure 4 shows how WCR and spray area vary with the number of sprayer nozzles across the set of detectors. We also plotted the WCR and area sprayed against mAP, as shown in Figure 5.

V. DISCUSSION

Table II shows that Yolov5m has the highest mAP values for both sugar beet and weeds, on both the LB and BB datasets. The mAP for both classes in the LB dataset is lower than in BB partially due to the higher density and smaller size of the sugar beets and weeds in LB. A traditional ML evaluation would stop there. But recalling our motivation to devise a strategy that assesses the practical feasibility of ML methods for precision spraying applications, we need to examine the results in more depth. Table II shows that YoloV3 has the fastest inference speed. The trade-off here is to select YoloV5m and process 74.1 FPS very accurately or select YoloV3 and process 75.75 FPS, but less accurately.

Next, we look at the trade-offs that come with different batch sizes. Table II only reports results where batch size = 1, whereas Table III compares results for higher batch sizes. If images from 35 cameras were processed in parallel on the GTX 2080Ti (i.e. batchsize = 35), then a frame rate of 212 or 263 FPS (BB or LB, respectively) could be achieved with YoloV5m on a single GPU. However, the Tesla T4 only achieves a frame rate of 158 or 163 FPS. While results from both GPUs fall short of the speed required to spray at our target pace of 15mph (6.7m/s), a total of 4 and 6 GPUs would make in-the-field spraying with YoloV5m feasible using the GTX 2080Ti and Tesla T4 respectively.

Furthermore, we investigate the trade-offs when considering our WCR and area sprayed metrics. Figure 4 and Table IV show that increasing the number and precision of nozzles decreases both the area sprayed and WCR. This trend appears across datasets and detection models. Among detectors, Yolov5m, yolov3 and Faster RCNN offer the best

![Table II: Performance for object detectors with the BB (top) and LB (bottom) datasets. The fastest inference speeds (FPS) and highest mAP scores are highlighted.](image)

![Table III: Inference speed in frames per second (FPS) on a GTX 2080Ti and a Tesla T4 for object detectors with different batch sizes for the BB (top) and LB (bottom).](image)

As per Section III, that would be a frame rate of 980 or 1024, depending on the number of cameras.
TABLE IV: WCR and area sprayed for 1–4 nozzles, for BB (top) and LB (bottom). Highlighted values explained in text.

| Model      | Backbone | 1 nozzle | 2 nozzles | 3 nozzles | 4 nozzles |
|------------|----------|----------|-----------|-----------|-----------|
|            |          | Weed     | Area      | Weed      | Area      | Weed      | Area      | Weed      | Area      |
|            |          | coverage rate | sprayed   | coverage rate | sprayed   | coverage rate | sprayed   | coverage rate | sprayed   |
| yolov5m    | DN-53    | 99.84    | 64.26     | 94.81     | 25.97     | 85.78     | 15.34     | 74.72     | 11.35     |
| yolov4     | DN-53    | 99.84    | 82.48     | 93.4      | 45.6      | 81.72     | 30.47     | 70.08     | 23.9      |
| yolov3     | DN-53    | 99.84    | 64.51     | 94.34     | 26.05     | 84.68     | 15.27     | 75.4      | 11.19     |
| Faster R-CNN | R-50 | 99.03    | 67.31     | 94.52     | 28.34     | 85.78     | 17.08     | 75.08     | 12.51     |
| Faster R-CNN | R-101 | 98.86    | 66.53     | 93.06     | 27.77     | 85.21     | 16.28     | 73.47     | 12.04     |
| Faster R-CNN | Rx-101 | 99.68    | 67.61     | 93.87     | 28.18     | 84.57     | 16.67     | 73.79     | 12.11     |

Fig. 4: The number of nozzles plotted against WCR (left) and area sprayed (right) for BB (top row) and LB (bottom).

Fig. 5: mAP for weed items plotted against WCR (left) and area sprayed (right) with 3 nozzles for BB (top row) and LB (bottom).

results in WCR and sprayed-area terms. This demonstrates that there is a trade-off between WCR and area sprayed. Given a particular detector, the number of nozzles can be selected to further optimise the WCR or the area sprayed to the desired extent. To achieve a high WCR, where the vast majority of weeds are targeted, fewer, less precise nozzles would be required. However, this would increase area sprayed and thus herbicide wastage. In growing environments where there is a higher tolerance for weeds and the emphasis is on herbicide reduction, targeting the weeds more precisely by increasing the number of nozzles would be more suitable. For both datasets, using 3 nozzles seems to produce a significant reduction in area sprayed while retaining a good weed coverage rate—which is what we want: high WCR combined with low area sprayed means that we are hitting a large number of weeds while wasting less herbicide. The highest WCR on the BB dataset is achieved jointly by YoloV5m and Faster RCNN (R-50) at 85.78 for 3 nozzles. However, YoloV5m only sprays 15.34% compared to 17.08% for Faster RCNN. For the LB dataset with 3 nozzles, YoloV5m achieves the highest WCR of 96.6 and sprays 36.33% of the area.

Figure 5 illustrates the relationships between the traditional mAP results (from Table II) and the new WCR and area sprayed metrics. WCR is positively correlated with mAP demonstrating that mAP is still a reliable indicator to compare the performance of different detectors for use in weed spraying systems. However, area sprayed is not well correlated with the mAP for the weed class. For example, in LB, the detector with the best mAP for the weed class,

9237
YoloV5m (34.2 mAP), uses ~15% more herbicide than other approaches such as YoloV3 (33.6 mAP) and Faster R-CNN (23.6 mAP). Additionally, in BB, all the detectors spray a similar area despite the mAP varying between these detectors. This suggests that a higher mAP for the weed class does not necessarily imply less herbicide wastage.

One limitation of our results is that they were computed on non-contiguous images. This means the entire spray resolution needs to fit within one image. However, if our images were contiguous we could calculate the impact on WCR and area by allowing the spray to run over into the next image frame. Other limitations include the time taken for the sprayer to turn on and off, and the assumption of rectangular spray zones, whereas these are typically circular. Spray drift due to wind force and vehicle movement is also an influencing factor for precision targeting. Modelling these aspects will provide more accurate estimates of sprayer performance, and they are all challenges that we will address in future work.

### VI. CONCLUSION

This paper has illustrated the utility of three metrics in evaluating the feasibility of different state-of-the-art object detection methods applied to selective spraying. Our WCR metric demonstrates that our weed detection methods are able to achieve a hit rate similar to broadcast spraying. WCR can be used to verify that sufficient weed spraying performance has been retained, while optimising for other factors such as speed and power consumption, even if detection accuracy is reduced. The inference speed metric proves that the models considered are fast enough to detect and spray weeds on-the-fly. At the same time, our area sprayed metric highlights options that produce clear reductions in area sprayed and hence herbicide required. As with many multi-criteria optimisation problems, there is no single clear winner. Our new metrics help highlight the advantages and drawbacks of different approaches, demonstrating that when it comes to practical deployment, it’s not just about mAP.

In future, we will look to improve the frame rate achievable on a single GPU. Additionally, we will improve our WCR and area sprayed estimations by using contiguous images and take into account more properties of the sprayer, including area shape and nozzle response times. Lastly, we will use a prototype precision sprayer to measure WCR and area sprayed in real field environments.

### REFERENCES

[1] X. Wu, S. Aravecchia, P. Lottes, C. Stachniss, and C. Pradalier, “Robotic weed control using automated weed and crop classification,” Journal of Field Robotics, vol. 37, no. 2, pp. 322–340, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21938

[2] S. K. Mathiassen, T. Bak, S. Christensen, and P. Kudsk, “The effect of laser treatment as a weed control method,” Biosystems Engineering, vol. 95, no. 4, pp. 497–505, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511006002984

[3] V. Nguyen Thanh Le, B. Atpoei, and K. Alameh, “Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods,” Information Processing in Agriculture, vol. 6, no. 1, pp. 116–131, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214317318300258

[4] J. Gao, A. P. French, M. P. Pound, Y. He, T. P. Pridmore, and J. G. Pieters, “Deep convolutional neural networks for image-based convolvolus sepium detection in sugar beet fields,” Plant Methods, vol. 16, no. 1, pp. 1–12, 2020.

[5] X. Jin, J. Che, and Y. Chen, “Weed identification using deep learning and image processing in vegetable plantations,” IEEE Access, vol. 9, pp. 10940–10950, 2021.

[6] A. Milioio, P. Lottes, and C. Stachniss, “Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs,” arXiv preprint arXiv:1709.06764, 2018.

[7] P. Lottes and C. Stachniss, “Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, pp. 5155–5161.

[8] N. Hussain, A. A. Farooque, A. W. Schumann, F. Abbas, B. Acharya, A. McKenzie-Gospill, R. Barrett, H. Alzaal, Q. U. Zaman, and M. J. Cheema, “Application of deep learning to detect Lamb’s quarters (Chenopodium album L.) in potato fields of Atlantic Canada,” Computers and Electronics in Agriculture, vol. 182, p. 106040, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016869922000582

[9] N. Hussain, A. A. Farooque, A. W. Schumann, A. McKenzie-Gospill, T. Esau, F. Abbas, B. Acharya, and Q. Zaman, “Design and development of a smart variable rate sprayer using deep learning,” Remote Sensing, vol. 12, no. 24, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/24/4091

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in European conference on computer vision. Springer, 2014, pp. 740–755.

[11] O. Hill, “How to properly set up a crop sprayer,” Farmers Weekly, https://www.fwi.co.uk/arable/crop-management/video-how-to-properly-set-crop-sprayer, 16th February 2017, (accessed 13th September 2021).

[12] “Speed + Spraying: What growers need to know,” https://www.etsprayers.com/blog/2017/03/20/speed-spraying-growers-need-know/, March 2017, (accessed 13th September 2021).

[13] P. Soviany and R. T. Ionescu, “Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,” in 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE, 2018, pp. 209–214.

[14] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V. Laughing, tkianai, yxNONG, A. Hogan, lorenzomammama, AlexWang1900, J. Hajek, L. Diacou, Marc, Y. Kwon, oleg, wanghaoyang1016, Y. Defretin, A. Lobia, ml5ah, B. Milanko, B. Fineran, D. Kromov, D. Yiwei, Doug, Durgesh, and F. Ingham, “ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations,” Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4679653

[15] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” arXiv preprint arXiv:1506.01497, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.