Crop stem detection and tracking for precision hoeing using deep learning

Louis Lac, Jean-Pierre da Costa, Marc Donias, Barna Keresztes, Alain Bardet

Version of Record: https://www.sciencedirect.com/science/article/pii/S0168169921006232
Manuscript_736d97c0685d9f1e16bc6cec39dbe064

Crop Stem Detection and Tracking for Precision Hoeing Using Deep Learning

Louis Lac[a,b], Jean-Pierre Da Costa[a,b], Marc Donias[a,b], Barna Keresztes[a,b], Alain Bardet[c]

[a] *Univ. Bordeaux, IMS UMR 5218, F-33405 Talence, France*
[b] *CNRS, IMS UMR 5218, F-33405 Talence, France*
[c] *CTIFL, 28 Route des Nebouts, 24130 Prigonrieux, France*

Abstract

Developing alternatives to the chemical weeding process usually carried out in vegetable crop farming is necessary in order to reach a more sustainable agriculture. However, a precise mechanical weeding requires specific sensors and advanced computer vision algorithms to process crop and weed discrimination in real-time.

In this paper we propose an algorithm able to detect, locate, and track the stem position of crops in images which is suitable for precision actions in vegetable fields such as mechanical hoeing within crop rows. The algorithm is two-fold: (i) a deep neural network for object detection is first used to detect crop stems in individual RGB images and then (ii) an aggregation algorithm further refines the detections taking advantage of the temporal redundancy in consecutive frames.

We evaluated the pipeline on images of maize and bean crops at an early stage of development, acquired in field conditions with a camera embedded in an experimental mechanical weeding system. We reported F1-scores of respectively 94.74 % and 93.82 % with a location accuracy around 0.7 cm when compared with human annotation. Moreover, this pipeline can operate in real-time on an embedded computer consuming as little power as 30 W.

Keywords: precision agriculture, deep learning, neural network, object detection, tracking algorithm

1. Introduction

Vegetable market weighs around 1250 B$ worldwide and keeps increasing steadily at a rate of 2.4 % a year (Indexbox (2020)). In this context and to ensure sustainable yields, weeding of vegetable fields is required for almost all kinds of crops (van Heemst (1985)). Weeding is usually handled using herbicides, which are inexpensive and efficient but recent awareness and criticism about the negative impact of phytosanitary products on soils and wildlife (Torretta et al. (2018)) has pushed organic farming practices in the spotlight (Lamichhane et al. (2019)).

However, the solutions to weed eradication without chemical products are limited. Organic farms use a mix of manual and mechanical weeding (Sanbagavalli (2020)) which expensive and repetitive for workers. To address those issues new innovative solutions aim at automating the weeding process. Some reviews show that automatic inter-row weeding is both feasible and economically viable (Pedersen et al. (2006)) but intra-row weeding is still challenging as the space between crops is much lower and crop distribution in the row is not always predictable (Griepentrog et al. (2004)).

1.1. Related Work

Several methods have been developed for crop and/or weed detection in vegetable farms which differ in the complexity and number of sensors embedded in the system as well as in the embedded algorithms.

Concerning the acquisition system, some methods employ simple RGB or RGB-NIR (Near Infra-Red) sensors (Jeon et al. (2011); Montalvo et al. (2012); Lottes et al. (2017); Bah et al. (2018); Lottes et al. (2018)) while others use more advanced sensors such as multi-spectral or hyper-spectral cameras (Gerhards and Christensen (2003); Wendel and Underwood (2016)) or depth-camera (Gai et al. (2020)). The first solution is often preferred as it is less expensive and usually more suitable for real-time applications (Griepentrog et al. (2004)). Sensors are mostly embedded directly on the farming robot (Gerhards

and Christensen (2003); Jeon et al. (2011); Montalvo et al. (2012); Wendel and Underwood (2016); Lottes et al. (2018)) but can also be carried by Unmanned Aerial Vehicles (UAV) (Lottes et al. (2017); Bah et al. (2018)).

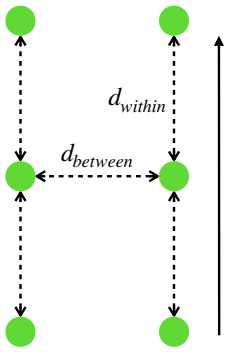
Regarding the crop and weed recognition task, semantic segmentation of images is almost always preferred, some only discriminate between crops and weeds while others also classify the species. Solutions that focus on inter-row hoeing of crops with high spacing often rely on standard computer vision methods to segment the image (Gerhards and Christensen (2003); Montalvo et al. (2012); Wendel and Underwood (2016); Lottes et al. (2017); Gai et al. (2020)). Local features and descriptors are often extracted using either radiometric indices, e.g. Excess Green Index (EGI), NDVI (Normalized Difference Vegetation Index) or geometrical and textural features, e.g. Fourier descriptors or other hand-tuned features descriptors. Classification is performed using methods such as Markov Fields, Principal Component Analysis (PCA), Random Forest or other machine learning classifiers. Other work methods take advantage of new deep learning frameworks such as semantic segmentation networks to perform classification (Jeon et al. (2011); Bah et al. (2018); Lottes et al. (2018); Wu et al. (2020)). Lottes et al. (2018) also takes into account the temporal aspect of the data.

1.2. The BIPBIP Project

The BIPBIP (Bloc-outil et Imagerie de Précision pour le Binage Intra-rang Précoce)[1] project aims at developing a precision weeding module based on fine mechanical hoeing that is designed to weed crops in the intra-row (illustrated in Figure 1a) without use of phytosanitary products. It is designed to weed one row at a time, but it can be replicated in parallel to operate on multiple rows at the same time within the same lane. Moreover, it is independent of its carrier and can be easily transferred to another vehicle. The module is built around two components: (i) a vision system that detects and tracks crop stems, further

[1]Tool-block and Precision Imaging for Early Intra-row Hoeing, `http://challenge-rose.fr/en/projet/bipbip-project/`

(a)



(b)

Figure 1: 1a: One BIPBIP weeding module embedded under an electric tractor operating on the left row of a maize bed in an experimental plot; 1b: Between-rows ($d_{between}$) and within-row spacing (d_{within}). Row direction is indicated by the arrow.

developed in section 2 and (ii) a mechanical weeding tool not addressed in this paper.

The module primarily targets market gardening (bean, onion, leek, etc.) but is also tested on field crops with large intra-row spacing (maize, sweet corn, rapeseed, etc.) as part of the ROSE Challenge organized by the French National Research Agency. Only stages of development between 2 and 5 weeks are considered as weed competition is at its highest during this growth period. However, currently only maize and bean are supported in the configuration described in Figure 1b with $d_{between}$ from 75 cm to 80 cm and $d_{within} = 15\ cm$ for maize and $d_{between}$ from 15 cm to 37.5 cm and d_{within} from 3 cm to 8 cm for bean. In this configuration weed infestation may be high, occlusions and obfuscations can occur, so the detection module should handle those edge cases correctly.

The mechanical weeding tool is currently composed of a metal tip that scraps the soil to remove all weeds without distinction around each crop of interest. A system not described in this paper can activate and move it along the row. In addition, two mobile plowshares placed on both sides of the module assist weeding in the inter-row. This system imposes a speed constraint for the overall

module of around 0.5 m/s. The advantage of such a system is that detecting weeds is not required for hoeing, only the crop stem positions need to be known as they are the only part of the crops to be avoided by the metal tip and the mobile plowshares during the weeding process.

1.3. Motivations and Contributions

This paper proposes a stem detection pipeline developed for the BIPBIP weeding module. This pipeline should operate in real-time and provide the stem position of crops with a great location accuracy which is required for the precision hoeing process. Our contribution is twofold:

- We propose a method to detect stem locations in images using an object detector, and we evaluate two alternatives: (i) approximating the stem location by the whole crop bounding box center, and (ii) detecting stems with a bounding box centered at its location.
- We propose a temporal aggregation algorithm which takes advantage of the temporal coherence of successive images to improve the detection performance of the network.

The article is organized as follows: the sensors and the database are presented in Section 2, the deep neural network and the aggregation algorithms are detailed in Section 3, results are described in Section 4 and conclusions are drawn in Section 5.

2. Materials

2.1. Vision System

The vision system illustrated in Figure 2 consists of a single 3 Mpixels industrial RGB camera (Basler acA2500-gc) equipped with a C125-0418-5M F1.8 f 4 mm Basler lens which can capture images at a rate of 15 frames per second (fps). It is pointing down with its principal optical axis perpendicular to the ground at a constant height of around 35 cm. This system is confined in a hull,

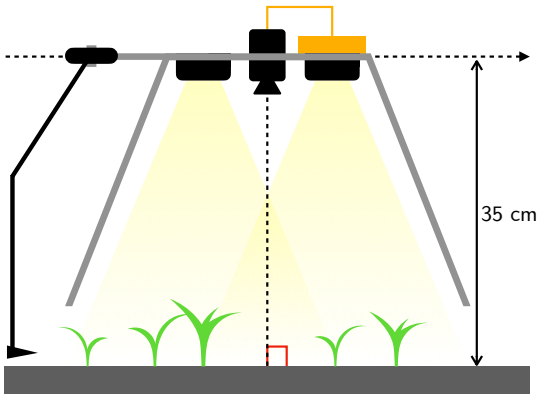


Figure 2: Schematic representation of the sagittal cross-section of BIPBIP hoeing system with the mechanical weeding tool (left), the vision system with the embedded computer (orange), the two LED panels and the camera (black). Forward direction is to the right.

sealing it from natural lighting. Light conditions are artificially controlled by two 20 W LED panels in order to obtain a brightness as uniform as possible. This setting avoids unpredictable conditions such as glare and overexposure and ensures a better robustness of the detection algorithm. Moreover, camera focus is set to match the camera height and the exposure and white balance are adjusted and fixed at the beginning of the weeding process.

The computation is processed in real time on an Nvidia Jetson Xavier which is an embedded computer optimized for deep learning and computer vision computations. Moreover, it can operate at the very low power consumption of 30 W max. The algorithms are developed in Python using Numpy and OpenCV, and the framework used for the neural network inference is written in C, C++ and CUDA.

2.2. Database

We acquired two databases[2] both for training and validation of our method. The acquisition is processed with the vision system described in Figure 2 embedded in a lightweight acquisition module that can be carried easily. Both

[2]The data is available on request to the author.

Figure 3: Samples of the image database used for the training and the validation of the object detector.

of them were captured in three locations in France: Montoldre in Auvergne-Rhône-Alpes, Liposthey in Nouvelle-Aquitaine (Fermes Larrère) and Lanxade in Nouvelle-Aquitaine (CTIFL).

The first one (that we call the image database) is an image collection used for training and validation of the deep learning algorithm presented in section 3.2. We used 80 % of the database for the training and the remaining 20 % for the evaluation. The images are either 3 Mpixels or 5 Mpixels and soil conditions diverge slightly: Nouvelle-Aquitaine soils are sandy while Montoldre ones are tougher and more dusty. This database currently supports three types of crops at an early stage of development (2 to 5 weeks): maize, bean and leek. However, as the leek database is currently not large enough to be representative, leek results are not presented in this paper. It is also designed to cover as many situations as possible such as weed infestation levels, soil types, grown stages, crop overlap and obfuscation, but it is continuously extended with new images of previously unseen conditions. Some samples are presented in Figure 3.

The second database (which we call the video database) is composed of four 15 fps videos saved as consecutive frames, two for maize crops and two for bean crops. It is designed to mimic the real acquisition context of the weeding module

Figure 4: Some samples of the four videos of the video database.

and is dedicated to the evaluation of the aggregation algorithm presented in section 3.3. Soil conditions and growth stages are different for each video, the first two of each crop contain crops at more advanced growth stages and packed tightly while the two last contain crops at an earlier stage. Some samples can be seen in Figure 4.

3. Methods

The developed pipeline is two-fold: (i) an object detection based deep neural network first provides stem locations in individual RGB images and then (ii) an aggregation algorithm further filters the detections, leveraging the temporal aspect of the successive frames.

Moreover, we compare two approaches for the stem detection part. In the first one the neural network is trained to detect entire crops and the stem is approximated by the crop bounding box center. In the second one the network is trained to directly detect the stem as an object. We believe that the second approach should give better location accuracy, but the network may struggle to learn these uncommon objects. In the following we use labels such as *Maize Crop* to denote the first configuration and *Maize Stem* to denote the second

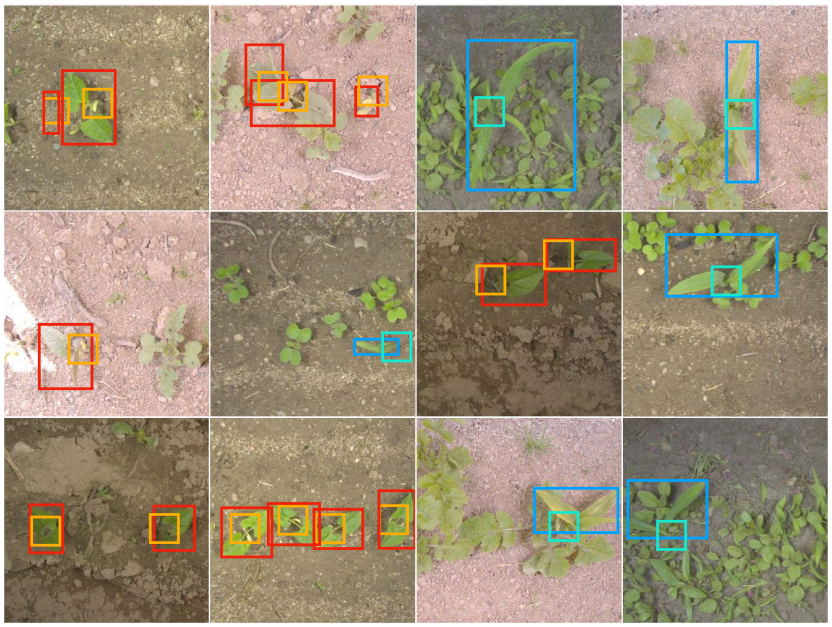


Figure 5: Samples of the image database with the annotations overlaid. Maize crops are annotated in blue and the stems in cyan, bean crops in red and the stems in orange.

configuration.

In the following, we first describe the database annotation process, then the two parts of the pipeline are detailed in Section 3.2 and Section 3.3.

3.1. Annotation

The databases presented in section 2.2 are annotated with bounding box ground-truths to provide labels for the neural network training and for the evaluation. The annotation work is performed with the labelImg[3] software, and we annotated both stems and entire crops for maize and bean. The crop bounding box is a rectangular box around the whole crop (red and blue boxes in Figure 5) while the stem bounding box is a square box centered on the stem entry point in the ground and with a side length normalized to be equal to 7.5 % of the image's smallest length (orange and cyan boxes in Figure 5).

[3] `https://github.com/tzutalin/labelImg`

Label	Images	Crop annotations	Stem annotations
Maize	1 034	2 095	2 133
Bean	748	2 820	2 824
Total	1 782	4 915	4 957

Table 1: Number of images and annotations for each type of crop.

For the image database introduced in section 2.2 we annotated all the images (statistics are shown in Table 1). As presented in section 4.1, it is further divided in two: one part for training (80 %) and the other one for validation (20 %).

The two maize videos are 1765 and 427 images long and 51 of them are annotated (144 crops). The two bean ones are 251 and 784 images long and 53 of them are annotated (263 crops). The annotation process is similar to the image database except that the annotation is not performed on every image to avoid tagging the same crop in multiple successive images, but every crop is annotated at least once.

3.2. Stem Detection with Neural Network

We propose to use an object detection neural network to regress stem locations. As our application requires real-time computation we chose a one-stage network over a two-stage because they achieve higher inference speed (Huang et al. (2017); Jiao et al. (2019)). In this family different designs are proposed, the most used being SSD (Liu et al. (2016), RetinaNet Lin et al. (2017)) and YOLO (Redmon et al. (2016)). In recent years SSD and RetinaNet were supplanted by more accurate and faster networks such as EfficientDet (Tan et al. (2019)) and ASFF (Liu et al. (2019)) while YOLO underwent several enhancement iterations (Redmon and Farhadi (2017, 2018)). Recently a team of researchers developed YOLOv4 (Bochkovskiy et al. (2020)) with the aim to achieve a high quality object detector which is simple to train and ready for production by aggregating the newest deep learning features that can improve the speed-accuracy trade-off. A simplified overview of the architecture is presented is Figure 6. We chose this framework as it is the most accurate and faster to our knowledge and is ready

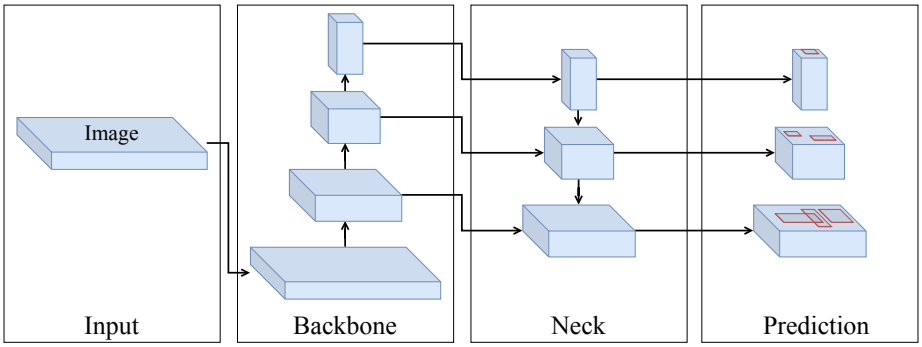


Figure 6: YOLOv4 object detector simplified architecture with CSPDarknet53 backbone (Wang et al. (2019)), PANet neck (Liu et al. (2018)) and YOLOv3 anchor-based prediction head (Redmon and Farhadi (2018)).

for production. This framework still offers a wide variety of networks achieving different speed-accuracy trade-offs ranging from Tiny YOLOv4 –a smaller variant that can run faster– to YOLOv4 –a more accurate but slower variant.

During inference, we extract the bounding box centers, which represent the crop stem locations in our application. In the following, the stem detections for image I_n are noted D_n.

In section 4 we benchmark some variants to highlight the speed-accuracy trade-offs and to choose a variant suitable for our application.

3.3. Temporal Aggregation

Object detection operates image by image, but the weeding robot requires a unified detection in order to make a decision. For this purpose we can take advantage of the temporal redundancy of detections caused by the overlap of successive images to improve the accuracy and provide better confidence indicators. We propose a simple approach where stem detections from different images are first projected in a common referential via the computation of the optical flow, then they are aggregated to remove duplicates and recover missed detections.

We formulated several hypotheses to simplify the problem, (i) the ground and crops are assumed rigid bodies with no relative displacements between any

part of them, (ii) the ground is assumed to be planar, (iii) the camera principal axis is perpendicular to the ground and (iv) the camera displacement is in the horizontal plane i.e. no changes of height. With these hypotheses in place the optical flow of the soil pixels between two consecutive images is assumed to be constant and equal at every pixel location, and it can be used to map ground points in an image to another image knowing that displacement. As we defined the stems as the entry point in the soil, stems from one image can be mapped to another one knowing the ground displacement between them.

We propose an iterative algorithm which operates at each new image. It takes as input the new image I_n and the past one I_{n-1}, as well as the stem detections D_n of image I_n. It updates two pieces of data: $\Delta d_n|_{\mathcal{R}_1} \in \mathbb{R}^2$ which is the total translation of the frame of reference $\mathcal{R}_n$ associated to the image I_n to the first image one $\mathcal{R}_1$, and a set of aggregated stem detections T_n where each detection is a location $p \in \mathbb{R}^2$ expressed in the frame of reference $\mathcal{R}_1$ (the initial conditions are respectively zero and the empty set). The referential $\mathcal{R}_n$ with axes (X_n, Y_n) associated to the image I_n has an origin located at the top-left image corner as illustrated in Figure 8.

The algorithm is composed of three parts illustrated in Figure 7: (i) the soil mask extraction (in blue) extracts the pixels belonging to the soil, (ii) the displacement computation (in orange) computes the translation from the first image to the current one and (iii) the aggregation (in turquoise) projects the stem detections in $\mathcal{R}_1$ and aggregates them.

3.3.1. Soil Mask Extraction

The soil mask extraction algorithm (in blue in Figure 7) computes a binary mask $M_n \in \mathbb{N}_{[0,1]}^{W \times H}$ where W and H are the image width and height, which means that in M_n pixels of the soil class have a value of one. This mask is used in the displacement computation algorithm to compute the optical flow of the soil points only.

The first step is the Non-Vegetal Mask Extraction (NVME). The Excess Green Index (EGI) (Woebbecke et al. (1995)) of the image I_n is first computed.

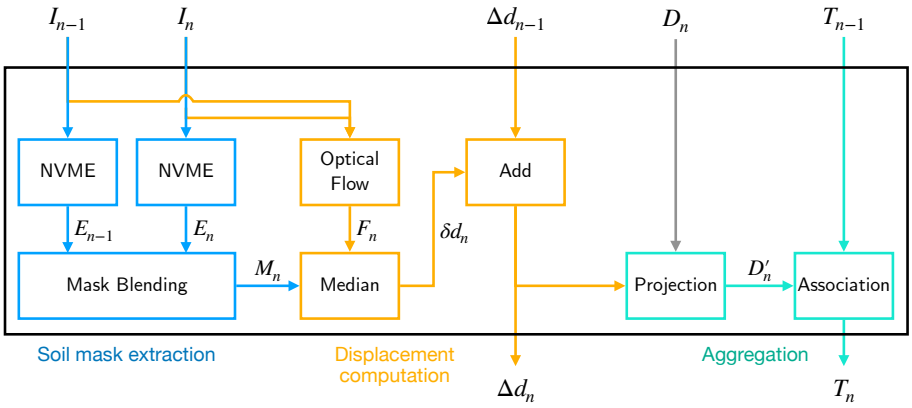


Figure 7: Overall scheme of the iterative aggregation process. Soil mask extraction (blue), displacement computation (yellow) and aggregation (turquoise). Mathematical notations are introduced in section 3.3 and reference frames have been omitted for clarity. NMVE stands for Non-Vegetal Mask Extraction.

A threshold $t_e \in \mathbb{N}$ is then applied to the EGI to obtain the mask of the soil:

$$E_n = \begin{cases} 1 & \text{if egi}(I_n) < t_e, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The same process is applied to I_{n-1} to obtain E_{n-1}.

The second step is the Mask Blending. This step combines the two successive masks E_n and E_{n-1} to extract pixels of soil class from both images. As the dense optical flow computed in the next section can be less precise at object boundaries and at pixel locations that are obfuscated in one of the images, this operation helps to remove those pixels from the mask.

$$E'_n = \begin{cases} 1 & \text{if } E_n = 1 \text{ and } E_{n-1} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

A second operation applies a morphological closing (morphclose) with a disk kernel having a radius of ten pixels to remove small holes and a binary mask of

the image safe area M_s (area without visible tool or hull parts) is applied:

$$M_n = \begin{cases} 1 & \text{if morphclose}(E'_n) = 1 \text{ and } M_s = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Illustrations of the intermediate soil masks and of the blended masks are shown in Figure 8a.

3.3.2. Displacement Computation

The displacement between the current image I_n and the first image I_1 is computed by integration of the displacement between successive images from I_1 to I_n. First, the dense Optical Flow $F_n|_{\mathcal{R}_n \to \mathcal{R}_{n-1}} \in \mathbb{R}^{W \times H \times 2}$ from I_1 to I_{n-1} is computed using Farnebäck's polynomial expansion (Farnebäck (2003)). The flow is then masked with M_n to gather the flow of the soil pixels only and the median value is calculated:

$$\delta d_n|_{\mathcal{R}_n \to \mathcal{R}_{n-1}} = \text{median}\left\{F_n(x,y)|_{\mathcal{R}_n \to \mathcal{R}_{n-1}} \setminus M_n(x,y) = 1\right\}. \tag{4}$$

This value represents the translation from $\mathcal{R}_n$ to $\mathcal{R}_{n-1}$ as a real value 2-dimensional vector.

The total translation from $\mathcal{R}_n$ to the first image reference frame $\mathcal{R}_1$ is then computed by summation with the previous total translation:

$$\Delta d_n|_{\mathcal{R}_1} = \Delta d_{n-1}|_{\mathcal{R}_1} + \delta d_n|_{\mathcal{R}_n \to \mathcal{R}_{n-1}}. \tag{5}$$

The relative translations and total translation are illustrated by the orange arrows in Figure 8c.

3.3.3. Aggregation

The aggregation process described in turquoise on Figure 7 is two-fold: (i) the detections D_n of image I_n are projected in the first image referential $\mathcal{R}_1$, then (ii) those detections are associated with previous ones representing the same stem object, resulting in a set of aggregated detections T_n that we called a "tracker".

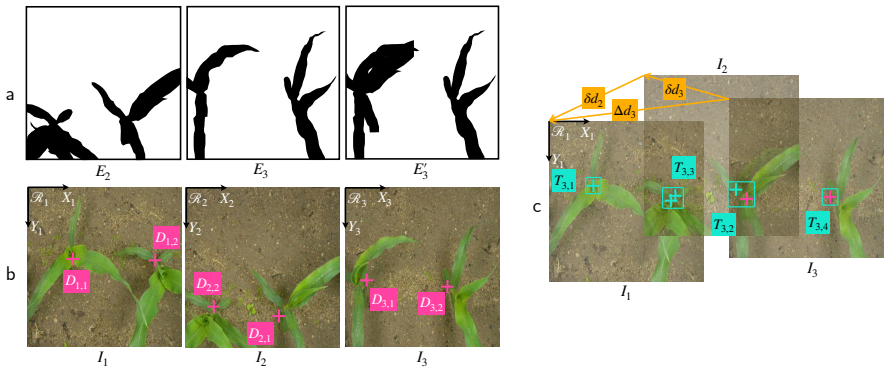


Figure 8: Simplified illustration of the aggregation process on three consecutive images. a: non-vegetation masks and blended soil mask. b: individual stem detections (pink) in their original reference frame. c: relative image translations and total displacement (orange), projected detections in the first image reference frame (pink) and trackers with past detections (turquoise). The frame-rate is artificially low for clarity.

The stem detections D_n are relative to the $\mathcal{R}_n$ reference frame of image I_n. The projected detections D'_n are first obtained by projection of D_n in $\mathcal{R}_1$ with an element-wise addition:

$$D'_n = D_n + \Delta d_n|_{\mathcal{R}_1}. \tag{6}$$

These projected detections are illustrated by pink crosses in Figure 8c.

In a second step, the projected detections of image I_n are associated with detections from previous images. We call a tracker a set of stem detections from different images associated through this aggregation process and representing the same underlying stem object. We define the canonical position of a tracker as the average position of its stem detections (isobarycenter). Moreover, a tracker has a lifetime. It is considered active if a detection has been associated with it during the last $max_{inactive} \in \mathbb{N}$ time steps, otherwise it is considered inactive. Only active trackers are considered during the association process while the aggregate detections T_n are composed of both active and inactive trackers.

The aggregation algorithm is based on the association process of the COCO AP evaluation metric (Lin et al. (2014)) and we use the Euclidean distance

Algorithm 1 "Merge" algorithm for detection-tracker association.

Inputs: T_{n-1} and D'_n
Parameters: $max_{inactive}$ and max_{dist}
$trackers \leftarrow$ `filterInactive`$(T_{n-1}, max_{inactive})$
$detections \leftarrow$ `sortByDecreasingConfidence`(D'_n)
$matches \leftarrow$ **new** `list`
for $detection \in detections$ **do**
  $(bestDistance, bestTracker) \leftarrow (+\infty, nil)$
  for $tracker \in trackers$ **do**
    $distance \leftarrow \|$`barycenter`$(tracker) - detection\|_2$
    if $distance < bestDistance$ **then**
      $(bestDistance, bestTracker) \leftarrow (distance, tracker)$
    end if
  end for
  if $bestDistance < maxDist$ **and** $bestTracker \notin matches$ **then**
    $matches \leftarrow matches \cup (detection, bestTracker)$
  end if
end for
return matches

between projected detections D'_n and tracker isobarycenters as the similarity metric for the association. The aggregation process is detailed in Algorithm 1. The detections D'_n are first sorted by decreasing confidence. Then, for each detection the Euclidean distance to every active tracker T_{n-1} is computed. If the closest tracker is at a distance below a threshold $maxDist \in \mathbb{R}$ and if it has not been previously associated with another detection, the detection is added to that tracker. This process results in an updated set of trackers T_n.

At inference and evaluation time, a tracker is considered valid if the number of detections it represents is above a threshold $minDets \in \mathbb{N}$. In section 4 we evaluate the influence of the distance threshold $maxDist$ and detection number threshold $minDets$.

Trackers are illustrated in Figure 8c by turquoise bounding boxes.

4. Results

We evaluated the two components of our system –object detection and aggregation– independently. Firstly, in Section 4.1 we evaluate different YOLO architectures based on standard object detections metrics, and we select the one that best suits our needs for the following evaluation. Secondly, in Section 4.2 we evaluate the temporal aggregation algorithm with a custom metric designed to assess the stem detection ability. We compare the two approaches described in Section 3, i.e. the whole crop bounding box v.s. the stem bounding box, and we perform a grid-search on two main hyper-parameters of the temporal aggregation algorithm. Finally, we discuss the results in Section 4.3.

4.1. Stem Detection

We compare 3 networks of different depth and structure: (i) YOLOv4 is the deepest network thus potentially the most accurate but also the slower, (ii) Tiny YOLOv4 (YOLOv4 T) which is a shallow variant of YOLOv4 expected to be faster and (iii) Tiny YOLOv3 3L (YOLOv3 T3L) which is a former tiny YOLO variant trained for comparison purposes.

We evaluate the performance with the standard COCO object detection metrics (Lin et al. (2014)). More specifically, we use the $AP_{0.5:0.95}$ (AP), the AP_{50}, AP_{75} and the AR_{100}. Moreover, we provide the mean Intersection over Union[4] (mIoU), which is computed at a 50 % IoU threshold and a fixed confidence threshold of 80 % for YOLOv4 and 25 % for Tiny variants (found via a grid-search not presented in this paper). We also report the inference speed in frames per second (FPS).

We trained the networks to detect 6 different classes at once: maize, bean and leek crops and their stems on the image database presented in section 2.2. In this paper we focus on maize and bean, so only those results are reported. We split the image database in a training set and a validation set with an 80 %-20 %

[4]The IoU is also known are the Jaccard index which is a measure of similarity between sets.

Network	AP	AP_{50}	AP_{75}	AR_{100}	mIoU	FPS
YOLOv4	53.87	89.71	54.59	61.20	80.96	13
YOLOv4 T	47.28	86.37	44.79	55.36	78.10	95
YOLOv3 T3L	38.77	82.31	31.64	48.62	75.44	90

Table 2: Object detection performance (in percent) and inference speed (fps) on the NVIDIA Jetson Xavier including video acquisition and post-processing for the three evaluated networks.

ratio and trained for 10,000 iterations of batches of size 64 images. We also used transfer learning (Athanasiadis et al. (2018)) from networks pre-trained on ImageNet (Deng et al. (2009)). The database is also augmented to reduce overfitting with the following transformations: random image scaling (from x0.4 to x1.6 the original size), random color changes (hue ±10 %, saturation and exposure from 1 to 1.5) and image Mixup (Zhang et al. (2017)). For practical reasons we chose an input size of 544 × 544. A higher input size would yield lower training and inference speeds and more working memory, and a lower size would degrade the accuracy too much. Preliminary experiments not listed in this paper showed that this input resolution achieves a suitable trade-off for our application.

The training is performed on a dedicated workstation running Ubuntu 18 LTS with an Intel Core i7-7700 4 Cores at 3.6 GHz CPU 32 GB and an Nvidia GeForce RTX 2080 SUPER 8 GB GPU. This setting is sufficient to handle training for our current database in approximately 2 hours for small networks (Tiny YOLOv4) to 7 hours for the largest one (YOLOv4).

Table 2 shows that there is a clear trade-off between object detection accuracy and the inference speed that can be obtained. On the Nvidia Jetson Xavier, Tiny YOLOv4 is more than 7 times faster than YOLOv4 at the cost of 6.59 % AP. YOLOv4 yields a 3.34 % higher AP_{50} and an mIoU 2.86 % higher. YOLOv3 T3L is both slower (-5 fps i.e. -5 %), less accurate (-8.51 % AP) and less precise in bounding box regression (-2.66 % mIoU) compared to Tiny YOLOv4, which illustrates the performance gains introduced with newer YOLOv4 networks. The AR_{100} illustrates the same trend.

Network	AP_{50}				AP			
	MC	BC	MS	BS	MC	BC	MS	BS
YOLOv4	94.47	95.07	90.60	78.69	69.06	71.11	41.84	33.49
YOLOv4 T	93.01	95.07	82.63	74.76	60.00	64.18	35.53	29.42
YOLOv3 T3L	92.02	91.50	77.59	68.13	51.01	51.82	28.09	24.13

Table 3: Object detection performance by object class (%). MC: whole maize crop, BC: whole bean crop, MS: maize stem, BS: bean stem.

When comparing AP and AP_{50} by crop type in Table 3 it appears that all networks have more difficulties in detecting the stems compared to the entire crop. For instance Tiny YOLOv4 loses 10.38 % AP_{50} for maize stems and 20.31 % AP_{50} for bean stems when compared to the whole crop (and respectively 29.53 % and 34.76 % for the AP). The drop in performance can be explained by two aspects: (i) the standard object detection metrics may not be suitable to evaluate the stems in the way we defined them as objects rather than keypoints, (ii) stems are more difficult to detect because of their small size, their less well-defined boundaries and the higher obfuscation they may suffer from. Moreover, it can be observed that maize stems are better detected than bean stems. This difference can be explained by the crop layout for bean crops that is narrower than the maize one (cf Figure 1b), thus leading to more obfuscation and overlap.

These results made us choose Tiny YOLOv4 for our application. It is fast enough for real-time use while still leaving some GPU time for other algorithms and its accuracy is sufficient, and could be improved in the future by increasing the input resolution for instance.

4.2. Stem Aggregation

We chose to evaluate the stem aggregation algorithm with a metric that better models our algorithms and their application. While the COCO AP is suitable for object detection evaluation, it is not for keypoint evaluation. Thus, we replaced the IoU similarity metric of the AP by the Euclidean distance be-

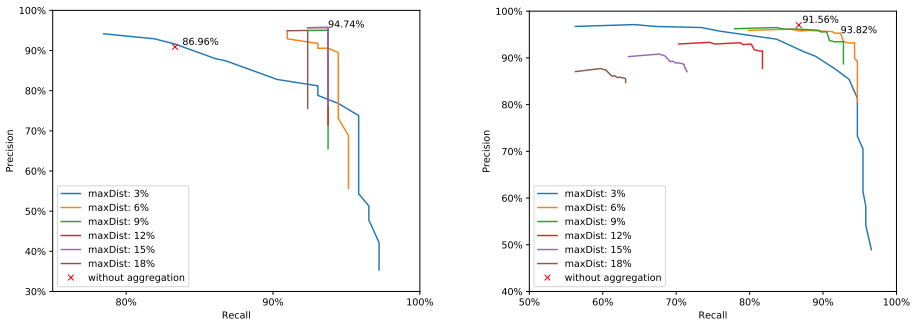
tween reference stems and predictions and the mean IoU is replaced by Location Accuracy (LAcc). Moreover, contrary to the AP we chose to fix the confidence threshold of the detector to its optimal value (25 % for Tiny YOLOv4) and the distance threshold (which is analog to the IoU threshold) is also fixed to 2 cm, which is a value suitable for the precision agriculture task targeted. This allows the evaluation of the precision, the recall, F1-score (Olson and Delen (2008)) and Location Accuracy. We believe that this metric is more concrete than the AP and measures more directly the hoeing performance (potential crop losses and false alarms). This evaluation is performed on the video database presented in section 2.2.

We chose a value of $max_{inactive} = 30$ for the tracking algorithm as this parameter is constrained by the speed of the hoeing module and the camera frame-rate. We also chose a value of $t_e = 40$ by a qualitative observation of the generated soil masks.

In the following, we present three experiments: (i) as the aggregation algorithm depends on two hyper-parameters –namely $minDets$ and $maxDist$– we produced a grid-search using the F1-score as the comparison metric to find the local optimum of those values for the stem detection task, (ii) we evaluated the performance of the aggregation algorithm and (iii) we compared the two approaches described in section 3 which are either using the crop bounding box or using the stem bounding box to regress the stem position.

4.2.1. Grid Search

We performed a grid-search for the two crop stems –maize stem and bean stem– independently in the following configuration: (i) $minDets$ varies from 1 to 20 by 1 increment, (ii) $maxDist$ varies from 3 % to 18 % by 3 % increment (expressed as a percent of the image's smallest side length). We plotted the precision-recall curves with respect to the combinations of those two parameters. On Figure 9 each curve varies along the $minDets$ parameter and the $maxDist$ parameter variation is presented by different curve colors. We chose the best combination of the two parameters based on the F1-score at that point in the



(a) Maize

(b) Bean

Figure 9: Precision-recall curves of the temporal aggregation performance on maize and bean stems with respect to the *minDets* parameter (curve line) and *maxDist* (curve color). *minDets* ranges from 1 detection to 20 detections. The F1-score for the best combination of these parameters are highlighted and the F1-score of the stem detection without the aggregation algorithm is shown as a red cross.

precision-recall curve. This value is shown in Figure 9 as well as the F1-score without the aggregation algorithm (red cross) for comparison.

Globally for the two grid-search presented in Figure 9 it can be observed that there is a local optimum for the *maxDist* parameter, in ascending order the curves first get closer to the top-right corner (which represents the best possible F1-score) and then move away from it. The same behavior is observed with the *minDets* parameter, a low value gives an excellent recall but a poor precision (right end of the curves) and vice-versa (left side of the curves); and in-between an optimum value is attained. The optimum F1-score of 94.74 % is obtained with $minDets = 10$ detections and $maxDist = 12$ % (4.6 cm) for maize stems and the optimum F1-score of 93.82 % with $minDets = 13$ detections and $maxDist = 6$ % (2.3 cm) for bean stems. The lower *maxDist* value for bean can be explained by the wider crop spacing for maize that results in less confusion between adjacent crops, thus the constraint on this parameter can be relaxed.

Configuration	Recall	Precision	F1-score	LAcc
Maize Crop	26.39 %	27.94 %	27.14 %	1.25 ± 0.07 cm
Maize Stem	83.33 %	90.91 %	86.96 %	0.46 ± 0.03 cm
Aggregated Maize Stem	93.73 %	95.74 %	94.74 %	0.68 ± 0.03 cm
Bean Crop	88.59 %	94.33 %	91.37 %	0.64 ± 0.03 cm
Bean Stem	86.69 %	97.02 %	91.57 %	0.46 ± 0.02 cm
Aggregated Bean Stem	92.40 %	95.29 %	93.82 %	0.54 ± 0.02 cm

Table 4: Performance of the stem detection on the video dataset with and without the aggregation algorithm as well as the performance of the whole crop detection without aggregation. Standard errors for LAcc are provided. For the aggregation algorithm the optimal values for $minDets$ and $maxDist$ parameters found with the grid-search are used.

4.2.2. Temporal Aggregation Performance

Table 4 shows that the temporal aggregation algorithm improves the performance of the detection (e.g. the "Maize Stem" configuration) compared to the non-aggregated case (e.g. the "Maize Stem Aggr" configuration). At the optimal parameter values fixed in the previous section, the F1-score for maize stems is improved by +7.78 % and by 1.25 % for bean stems. The improvement is much higher for maize stems and the aggregation algorithm improves both the recall (+10.40 %) and the precision (+4.83 %) while for bean stems the recall is better (5.71 %) but the precision is lower (-1.73 %). We believe that this contrast can be explained by two factors: (i) as presented in section 4.1 the AP for bean stems is lower than the one of maize stems, thus the aggregation algorithm proceeds on lower quality detections ; and (ii) the crop layout for bean crops is tighter than the maize crop one, making the tracking –which is based on a distance metric– of bean stems less robust to erroneous associations during the "Association" step presented in section 3.3.

While the aggregation algorithm improves the detection performance it also slightly decreases the location accuracy of the detections. The maize stems location accuracy loses 0.22 cm and the bean stems location accuracy is lower by 0.08 cm. However, the location errors are low (6.1 mm on average) and

Figure 10: Examples of dispersion ellipses containing 1 sigma (i.e. 68 %) of the samples for maize (top row) and bean (bottom row). The first two columns correspond to the first video and the last two to the second video.

suitable for precision hoeing in both cases. The standard errors of the location accuracies do not show that the difference in performance is significant.

To give a better insight of the location uncertainty and detection error, dispersion ellipses (Saporta and Hatabian (1986)) are computed and illustrated in Figure 10. The ellipses are larger for big crops with many leaves (e.g. maize crops on the left of the top row) and for crops obfuscated by weeds (e.g. bean crops on the left of the bottom row), which indicate that the overall performance is lower in these more difficult cases. This figure also highlights one common location mistake done by the neural network which tends to detect the top of the crop when the stem entry point in the ground is hard to detect (obfuscation by leaves or weeds for instance). Due to the strong parallax introduced by our acquisition system this creates line patterns in a tracker's detections (green maize stem tracker in the first maize image and bean stems in the two last bean images).

Concerning the inference speed of the temporal aggregation algorithm, the latency is dominated by the computation of the optical flow which runs at around 15 fps while the aggregation in itself runs at more than 100 fps.

4.2.3. Crop vs Stem Detection

Table 4 shows that using a bounding box centered on the crop stem (e.g. "Maize Stem" configuration) yield better performance than detecting the whole crop and using the bounding box center as an approximation of the stem location (e.g. "Maize Crop" configuration). The F1-score is dramatically improved for maize crops (+59.82 %) and slightly improved for bean crops (+0.20 %). Also, for both crops the Location Accuracy is better when using the stem bounding box rather than the crop bounding box: -0.79 cm for maize stems and -0.18 cm for bean stems. The difference in performance between the two kinds of crop can be explained by the difference in their size: bean crops are generally much smaller than maize crops, thus the bean crop bounding box center is a better approximation of the stem location than the maize one. It can be noted that the bean crop configuration yields a better recall than the bean stem configuration (+1.90 %), which illustrates the better detection of whole crops compared to stem bounding boxes pointed out in section 4.1.

4.3. Discussion

We have shown that the proposed detection pipeline yields performances suitable for the targeted precision hoeing application, both in terms of detection (average F1-score of 94.28 %), location accuracy (average LAcc of 6.1 mm) and inference speed. The use of an off-the-shelf object detector to detect stem bounding boxes seems relevant as it yields a great accuracy and the implementation is well optimized for real-time applications. We note, however, that work such as (Verucchi et al. (2020)) allows even higher inference speed on specialized embedded systems such as the NVIDIA Jetson Xavier, permitting the use of deeper networks like YOLOv4 to improve the accuracy even further without slowing down other algorithms.

The developed temporal aggregation algorithm improves the accuracy of the detection, on average there are fewer missed stems and fewer false positives. However, it does not improve the location accuracy. Though less than 2 mm in average which seems not significant, the explanation for this decrease needs

further exploration to find the potential causes. We believe that the many hypotheses we stated about the hoeing module posture and soil model can be the cause of biases and noises in the aggregation process. For instance, these modeling errors could cause a dispersion or a drift of the aggregated stem detections, leading to a degradation of the location accuracy.

Our results also highlights the difference in performance between crop types. Maize stems are better detected than bean stems no matter the configuration of the neural network. They also benefit more from the temporal aggregation accuracy boost. One possible explanation is the tighter crop layout of bean crops compared to maize crops which may generate more overlap between adjacent crops and more uncertainty on the precise stem location.

Comparing our work to the available literature is challenging as there are few public databases available, and to our knowledge there is no comparable hoeing method or public dataset matching our application needs. Future work will focus on the publication of our dataset to remedy this situation. An area of improvement is the addition of more crop types, growth stages and soil variability to assess the robustness of our algorithms.

5. Conclusions

In this paper we propose a computer vision pipeline able to detect in real-time the precise location of crop stems which can be used in challenging precision agriculture tasks such as mechanical hoeing of the intra-row. The developed method is two-fold: (i) an object detector based neural network is first used to detect stems in RGB images and then (ii) an aggregation algorithm is used to further refine the detections by leveraging the temporal nature of the successive frames. We measured the efficiency of our algorithms on our database composed of maize and bean crops in two configurations: (i) stems as crop bounding box centers and (ii) stems as objects.

We evaluated the algorithms with the F1-score as well as a location accuracy metric and reported the best results using the small variant of Yolo4 named

Yolo4 Tiny in the configuration of stems as objects. Currently, the system can detect maize and bean stems with an F1-score of respectively 94.74 % and 93.82 % and a location accuracy of 0.7 cm and 0.5 cm, which is suitable for precision hoeing.

Future work will focus on key-points based neural networks (Zhou et al. (2019)) that may be best suited as our goal is to detect key-points rather than bounding boxes. Those networks are less common and require more work in order to obtain a stable training and a suitable inference speed. Additionally, our current execution speed is sufficient but recent work (Verucchi et al. (2020)) showed impressive results in optimizing execution speed of neural networks on specialized hardware which we can take advantage of to further improve inference speed or the power consumption of our system. Lastly we are planning on extending the current database with more images covering more conditions and more crop types.

6. Acknowledgments

We acknowledge the French Research Agency (ANR) for funding (grant ANR-17-ROSE-0001 - BIPBIP) and thank the organizers of the ROSE Challenge and all the partners of the BIPBIP project.

References

Athanasiadis, I., Mousouliotis, P., Petrou, L., 2018. A Framework of Transfer Learning in Object Detection for Embedded Systems. arXiv:1811.04863 [cs] arXiv:1811.04863.

Bah, M., Hafiane, A., Canals, R., 2018. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. Remote Sensing 10, 1690. doi:10.3390/rs10111690.

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs, eess] arXiv:2004.10934.

Deng, J., Dong, W., Socher, R., Li, L.J., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL. pp. 248–255. doi:10.1109/CVPR.2009.5206848.

Farnebäck, G., 2003. Two-Frame Motion Estimation Based on Polynomial Expansion, in: Goos, G., Hartmanis, J., van Leeuwen, J., Bigun, J., Gustavsson, T. (Eds.), Image Analysis. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 2749, pp. 363–370. doi:10.1007/3-540-45103-X_50.

Gai, J., Tang, L., Steward, B.L., 2020. Automated crop plant detection based on the fusion of color and depth images for robotic weed control. Journal of Field Robotics 37, 35–52. doi:10.1002/rob.21897.

Gerhards, R., Christensen, S., 2003. Real-time weed detection, decision making and patch spraying in maize, sugarbeet, winter wheat and winter barley. Weed Research 43, 385–392. doi:10.1046/j.1365-3180.2003.00349.x.

Griepentrog, H.W., Christensen, S., Søgaard, H., Nørremark, M., Lund, I., Graglia, E., 2004. Robotic Weeding, in: AgEng 2004, pp. 12–16.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3296–3297. doi:10.1109/CVPR.2017.351.

Indexbox, 2020. World - Vegetable - Market Analysis, Forecast, Size, Trends and Insights. Technical Report. IndexBox Inc.

Jeon, H.Y., Tian, L.F., Zhu, H., 2011. Robust Crop and Weed Segmentation under Uncontrolled Outdoor Illumination. Sensors 11, 6270–6283. doi:10.3390/s110606270.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R., 2019. A Survey of Deep Learning-Based Object Detection. IEEE Access 7, 128837–128868. doi:10.1109/ACCESS.2019.2939201.

Lamichhane, J.R., Messéan, A., Ricci, P., 2019. Chapter Two - Research and innovation priorities as defined by the Ecophyto plan to address current crop protection transformation challenges in France, in: Sparks, D.L. (Ed.), Advances in Agronomy. Academic Press. volume 154, pp. 81–152. doi:10.1016/bs.agron.2018.11.003.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal Loss for Dense Object Detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007. doi:10.1109/ICCV.2017.324.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context, in: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014. Springer International Publishing, Cham. volume 8693, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

Liu, S., Huang, D., Wang, Y., 2019. Learning Spatial Fusion for Single-Shot Object Detection. arXiv:1911.09516 [cs] arXiv:1911.09516.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path Aggregation Network for Instance Segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759–8768. doi:10.1109/CVPR.2018.00913.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single Shot MultiBox Detector, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham. pp. 21–37. doi:10.1007/978-3-319-46448-0_2.

Lottes, P., Behley, J., Milioto, A., Stachniss, C., 2018. Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection

in Precision Farming. IEEE Robotics and Automation Letters 3, 2870–2877. doi:10.1109/LRA.2018.2846289.

Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., Stachniss, C., 2017. UAV-based crop and weed classification for smart farming, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Singapore, Singapore. pp. 3024–3031. doi:10.1109/ICRA.2017.7989347.

Montalvo, M., Pajares, G., Guerrero, J., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J., Cruz, J., 2012. Automatic detection of crop rows in maize fields with high weeds pressure. Expert Systems with Applications 39, 11889–11897. doi:10.1016/j.eswa.2012.02.117.

Olson, D.L., Delen, D., 2008. Advanced Data Mining Techniques. Springer-Verlag, Berlin Heidelberg. doi:10.1007/978-3-540-76917-0.

Pedersen, S.M., Fountas, S., Have, H., Blackmore, B.S., 2006. Agricultural robots—system analysis and economic feasibility. Precision Agriculture 7, 295–308. doi:10.1007/s11119-006-9014-9.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. doi:10.1109/CVPR.2016.91.

Redmon, J., Farhadi, A., 2017. YOLO9000: Better, Faster, Stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. doi:10.1109/CVPR.2017.690.

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs] arXiv:1804.02767.

Sanbagavalli, S., 2020. Eco-friendly weed management options for organic farming: A review. The Pharma Innovation Journal , 4.

Saporta, G., Hatabian, G., 1986. Régions de confiance en analyse factorielle, in: Fourth International Symposium on Data Analysis and Informatics, Elsevier Science Publishers. pp. 499–508.

Tan, M., Pang, R., Le, Q.V., 2019. EfficientDet: Scalable and Efficient Object Detection. arXiv:1911.09070 [cs, eess] arXiv:1911.09070.

Torretta, V., Katsoyiannis, I., Viotti, P., Rada, E., 2018. Critical Review of the Effects of Glyphosate Exposure to the Environment and Humans through the Food Supply Chain. Sustainability 10, 950. doi:10.3390/su10040950.

van Heemst, H., 1985. The influence of weed competition on crop yield. Agricultural Systems 18, 81–93. doi:10.1016/0308-521X(85)90047-2.

Verucchi, M., Brilli, G., Sapienza, D., Verasani, M., Arena, M., Gatti, F., Capotondi, A., Cavicchioli, R., Bertogna, M., Solieri, M., 2020. A Systematic Assessment of Embedded Neural Networks for Object Detection, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 937–944. doi:10.1109/ETFA46521.2020.9212130.

Wang, C.Y., Liao, H.Y.M., Yeh, I.H., Wu, Y.H., Chen, P.Y., Hsieh, J.W., 2019. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. arXiv:1911.11929 [cs] arXiv:1911.11929.

Wendel, A., Underwood, J., 2016. Self-supervised weed detection in vegetable crops using ground based hyperspectral imaging, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Stockholm, Sweden. pp. 5128–5135. doi:10.1109/ICRA.2016.7487717.

Woebbecke, D.M., G. E. Meyer, K. Von Bargen, D. A. Mortensen, 1995. Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. Transactions of the ASAE 38, 259–269. doi:10.13031/2013.27838.

Wu, X., Aravecchia, S., Lottes, P., Stachniss, C., Pradalier, C., 2020. Robotic weed control using automated weed and crop classification. Journal of Field Robotics 37, 322–340. doi:10.1002/rob.21938.

Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D., 2017. Mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412 [cs, stat] arXiv:1710.09412.

Zhou, X., Wang, D., Krähenbühl, P., 2019. Objects as Points. arXiv:1904.07850 [cs] arXiv:1904.07850.