Artificial Intelligence in Agriculture 13 (2024) 18–31

Contents lists available at ScienceDirect

Artificial Intelligence in Agriculture

journal homepage: http://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/

Estimation of flea beetle damage in the field using a multistage deep learning-based solution

Arantza Bereciartua-Pérez [a,*], María Monzón [b], Daniel Múgica [a], Greta De Both [c], Jeroen Baert [c], Brittany Hedges [d], Nicole Fox [d], Jone Echazarra [a], Ramón Navarra-Mestre [b]

[a] TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Tecnológico de Bizkaia, Astondo Bidea. Edificio 700, 48160 Derio, Bizkaia, Spain
[b] BASF SE, Speyererstrasse 2, 67117 Limburgerhof, Germany
[c] BASF Belgium (BBCC - Innovation Center Gent), Technologiepark-Zwijnaarde 101, 9052 Gent, Belgium
[d] BASF Canada Inc, 510, 28 Quarry Park Blvd SE, Calgary, Alberta T2C 4P5, Canada

ARTICLE INFO

Article history:
Received 19 September 2023
Received in revised form 20 November 2023
Accepted 4 June 2024
Available online 06 June 2024

Keywords:
Convolutional neural networks
Deep learning
Plant phenotyping
Damage estimation
Plant crop detection and identification

ABSTRACT

Estimation of damage in plants is a key issue for crop protection. Currently, experts in the field manually assess the plots. This is a time-consuming task that can be automated thanks to the latest technology in computer vision (CV). The use of image-based systems and recently deep learning-based systems have provided good results in several agricultural applications. These image-based applications outperform expert evaluation in controlled environments, and now they are being progressively included in non-controlled field applications.

A novel solution based on deep learning techniques in combination with image processing methods is proposed to tackle the estimate of plant damage in the field. The proposed solution is a two-stage algorithm. In a first stage, the single plants in the plots are detected by an object detection YOLO based model. Then a regression model is applied to estimate the damage of each individual plant. The solution has been developed and validated in oilseed rape plants to estimate the damage caused by flea beetle.

The crop detection model achieves a mean precision average of 91% with a mAP@0.50 of 0.99 and a mAP@0.95 of 0.91 for oilseed rape specifically. The regression model to estimate up to 60% of damage degree in single plants achieves a MAE of 7.11, and R2 of 0.46 in comparison with manual evaluations done plant by plant by experts. Models are deployed in a docker, and with a REST API communication protocol they can be inferred directly for images acquired in the field from a mobile device.

1. Introduction

Digitalization and automatization are common practice in diverse applications fields, such as industry, health, entertainment, or agriculture. In fact, a large part of the processes in the agricultural sector could be automated. Smart farming is the term coined to refer to the digitalization and automatization of some processes in the agricultural sector. In recent years, different researches have addressed diverse topics on automated actions for plant analysis by means of image processing and deep learning techniques, such as disease identification in images acquired by mobile devices (Fuentes et al., 2017) (Picon et al., 2019) (Argüeso et al., 2020) (Picon et al., 2022), damage estimation in plants in the greenhouse (Gómez-Zamanillo et al., 2023) or pest identification (Johannes et al., 2017) (Li et al., 2021a) and quantification (Bereciartua-Pérez et al., 2022a) (Bereciartua-Pérez et al., 2022b). A survey of Deep Learning applications in agriculture can be found in (Kamilaris and Prenafeta-Boldú, 2018) (Tian et al., 2020) (Zhang et al., 2020).

In this work, a new methodology has been proposed to automate manual assessments of plant damage caused by an insect. The methodology has been validated for oilseed rape plants affected by the damage caused by flea beetle. The methodology can be applied to any other crop as long as dedicated dataset is available. The field plots show an example of the crop. After seeding, oilseed rape plants emerge in the field, isolated (spaced out) or in bunches. Oilseed rape plants are in early growth stages, i.e., between the cotyledon stage and the maximum of 3–4 leaves. Every plant in the plot must be detected and segmented. Then, automated assessments must be done over single plant images. Deep learning-based techniques were used to perform both actions. Some

* Corresponding author.
E-mail address: aranzazu.bereciartua@tecnalia.com (A. Bereciartua-Pérez).

https://doi.org/10.1016/j.aiia.2024.06.001

Fig. 1. Examples of oilseed rape plants in different situations. A) oilseed rape with early damage; b) oilseed rape plant with high degree of damage; c) bunch of oilseed rape plants in different growing stages.

examples with isolated single plants and bunches of plants are shown in Fig. 1.

The real input image of the algorithm is images of certain regions of the field plots with oilseed rape plants covering two seeded rows (See Fig. 2).

Damage estimation by experts is standardized and well defined in these plants. The rating scale ranges between 0 and 100% (dead plant) and it is typically recorded in increments of 5%. The criteria for damage estimation are gathered in Figs. 3 and 4.

In case the plant has more than one cotyledon or leaf, the average value for them all is considered as the final value for the plant damage. Fig. 4 shows some examples.

In this work, a novel two-stage algorithm has been developed with two different deep learning-based models. Initially, an object detection model detects single plants in the images of the plots of this specific crop. Then, a regression model estimates the degree of damage of the delimited single plants (see Fig. 5). This methodology has provided satisfactory results.

The paper is organized as follows. After an introduction to the problem, the related work is described. The materials and methods used are explained in Section 3. A detailed description of the algorithm is detailed in Section 4. Section 5 gathers the results. Discussion of the results is made in Section 6 before ending with the conclusions.

2. Related work

As stated above, multiple processes in the agricultural sector could be automated. One of these processes is field crop monitoring, which requires the control of the plot and the crops that are growing therein. Traditionally, crop monitoring requires manual labour, and it is highly relevant for the control of the growing stages, the diseases, the appearance of weeds, the application of herbicides or other relevant issues. Identification of the different crops and the number of plants per species is necessary to control the growth progress in the plot.

Some works have been identified in the literature that detect and count plants. (Hosseiny et al., 2020) proposes using a Fast R-CNN network to detect and count corn plants in images acquired with drones. Plants appear in different growing stages, but they not seem to have a high overlap. The model performs with limitation in early stages of the plants, where detection is not feasible since few pixels represent every plant. The major contribution of this work is the review of state-of-the-art methods of plant detection and counting and object detection methods. In the presentation of the results, they compare the achieved metrics with the ones provided by other methods, although they use the Hausdorff distance, which is not really used in object detection problems. Mean average precision (mAP) is frequently used. Although the algorithm can detect plants at different growing stages, the input images do not contain highly overlapped samples. In fact, the performance of the model is limited due to the early stages of the crops, where detection is not feasible since only a few pixels represent each plant.

(Valente et al., 2020) shows a detailed overview over methods from 2012 on to detect vegetation, plants, and perform crop identification. The initial methods were based on classical image processing techniques by means of analysis of separate RGB planes and multispectral imagery, thresholds, and specific vegetation indexes, some of them very well known by the community, such as Normalized Difference Vegetation Index (NDVI) for remote observation and quantification of vegetation. In this work, they propose a method where acquisition conditions are well established in terms of flight height and resolution, and they estimate the average size in pixels of a plant in a specific growing stage. This is the way they address the detection of plants in bunches, which is always the bigger limitation of the existing methods. They prepared the data set with images of single plants and bunches of plants, where every image has a label with the number of plants contained. The proposed method initially detects the plants by Otsu thresholding and NDVI index. After, the detected regions are then released to the

Fig. 2. Example of plot with two rows of plants.

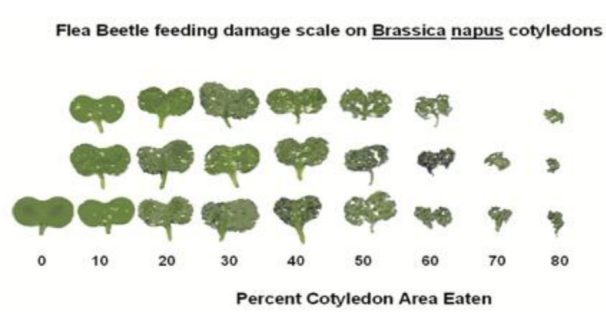


Fig. 3. Criteria for damage estimation according to cotyledon area eaten. "Strategies for Managing Flea Beetle Populations in Canada. Canola Council of Canada Project AG#2002-11 Year End Re". Province of Manitoba | agriculture - insect report 2013-06-12 (gov.mb.ca).

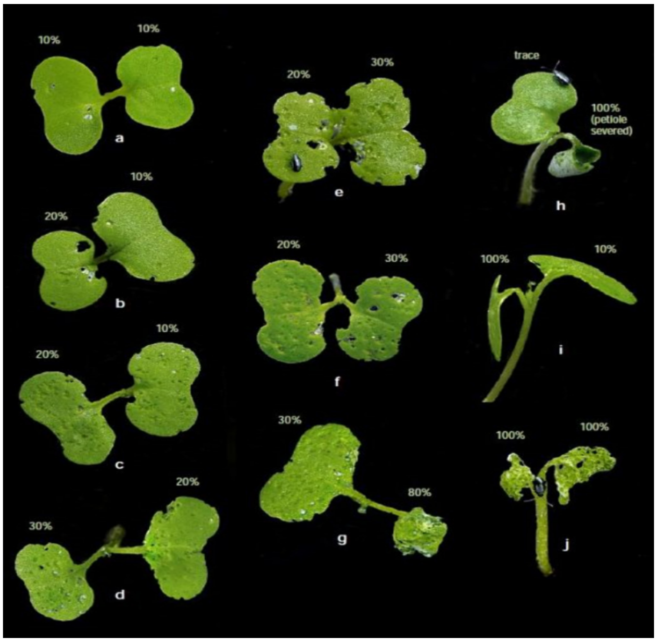


Fig. 4. Example of oilseed rape cotyledons and individual damage. Final plant damage is the average value. (Photo: Soroka & Underwood, AAFC-Saskatoon). Flea beetle – Prairie Pest Monitoring Network.

AlexNet based network, which has been trained to predict the number of plants contained in that image. Finally, post-processing action is added to provide the final number of plants. The splitting of the bunch is done equally, according to the estimated size per plant previously established. Therefore, this may lead to incorrect plant detection if individual plants are not equally distributed in the bunches. Furthermore, it seems difficult and prone to error to estimate the size of the plant since there are different stages of growth that can be different depending on the crop.

Other approaches for object counting that have been used in crowd counting or insects counting are based on density map estimations (Gao et al., 2020) (Bereciartua-Pérez et al., 2022a). These approaches do not detect the bounding box where the object is located but are focused on detecting the presence of the object and obtain the global number of elements. In counting works, these approaches outperform object detection methods, since a density map solution can tackle the problem of overlapped objects and partial views. Their aim is to count and not to locate with precision.

(David et al., 2021) propose a deep learning-based method where plants are detected in images acquired with drones. They use Fast R-CNN architecture and validate the results over images with three crops, maize, sunflower, and sugar beet. The images only contain one type of crop in every plot. The authors compare the results with those obtained by classical image processing approaches, and the deep learning approach outperforms classical methods. However, the method gets confused when the growing stage of the plants is such that the plants appear in bunches, overlapping each other. The images shown in the paper do not seem to have many overlapping plants.

IntegrateNet is a network that performs density map estimation to provide the final number of plants in the image. The model is trained and tested on maize plants, achieving very good results. This model does not perform location of single plants, and only maize plants are present in the images. It is not expected to find any other plant species (Liu et al., 2022).

In this recent work (Bai et al., 2022), the authors present a new deep learning network RiceNet that can perform rice plant counting, locating, and sizing in the field with high-throughput RGB images from drones. They propose a new network, named RiceNet, that generates density maps and combines them with attention map and size estimation head to properly locate single plants that show certain overlap.

Recent work in (Shi et al., 2022) proposes a new multibranch network capable of counting and locating plants. They add different branches, discriminator, and combination of losses to strengthen the extraction of meaningful features so as to make the model robust to tackle different domains in the dataset. They validated the method on three public datasets and achieved satisfactory results. Every dataset contains images of single crop.

What all of the aforementioned works have in common is that there are no multiple crops in the same image. Only a single crop is to be detected and counted. Our problem entails additional difficulty because in a non-controlled environment, multiple crops can appear simultaneously in the same image, although only damage over one crop will be evaluated. It is necessary to have a multiclass detecting solution.

Regarding the analysis of damage in plants, there are interesting reviews that compile some methods (Singh et al., 2016) (Singh et al., 2018) (Singh et al., 2020). Identification of the damage, either due to the disease or other causes, is usually carried out in the leaf and not in the complete plant, the total damage of the plant being the sum of the partial damage identified for each leaf. The mentioned methods range from classical machine learning and image processing techniques to recent developments based on Deep Learning. Future trends and prospects are also discussed in (Singh et al., 2018). This paper presents a review of the latest studies in this field, along with a comparison of

a b

Fig. 5. a) Example of image of dataset for object detection model; b) example of image of dataset for regression model for damage estimation.

the results obtained by the different techniques, the main problems and limitations encountered, and future trends.

A more recent review shows some methods that recognize plant leaf diseases, including conclusions and trends after comparing segmentation techniques, feature extraction, and popular classification methods (SVM, Random Forest, kNN). In recent years, solutions based on Deep Learning techniques have emerged and have been consolidated in the scientific community. Some works that evaluate diseases and possible damages have been identified, but these solutions are also restricted to the study of leaves (Agarwal et al., 2020; Jiang and Li, 2020; Johannes et al., 2017; Sai Reddy and Neeraja, 2022; Sharma et al., 2020). In fact, there are some reviews where different approaches are estimated to initially perform leaf segmentation and then disease detection and damage estimation (Saleem, 2019) (Li et al., 2021b).

There is an interesting approach to establish soybean plant damage with remarkable precision (Ghosal et al., 2018). In this work, the categorization of stress (damage) in the plant is due to several reasons, namely, bacteria, fungi, nutritional deficiency and herbicide damage. Nine stress categories were identified for the soybean plant: eight different stress situations and healthy leaf. The authors propose to solve the problem based on a classification approach with Deep Learning. The quantification of severity is also addressed. The segmentation of the soybean leaf is previously manually performed not to add noise. These are ideal conditions because no errors are added due to inaccurate leaf segmentation. Finally, just to point out that the riskiest works deal with images of leaves in nature, with a non-uniform background, lighting conditions and changing brightness (Picon et al., 2019) (Argüeso et al., 2019) (Ji et al., 2020). The severity of the disease is estimated in (Esgario et al., 2020), although it is not done quantitatively (percentage of the affected leaf) but one of the five possible qualitative stages of the disease is assigned to the leaf.

Other recent works have been identified that detect and classify the degree of disease in plants, mainly on leaves. This is the case of (Ahad et al., 2023) where nine different diseases of rice are distinguished. The authors propose six different networks and conclude that the best solution for disease estimation is the ensemble use of the networks. (Houetohossou et al., 2023) is an interesting review of recent methods based on deep learning approaches that tackle the problem of biotic and abiotic stress detection in fruits and vegetables. The authors propose an exhaustive work where they rank the most used CNN architectures, the used metrics, and the distribution of works around the world. This provides an interesting overview of the currently developed methods. (Shi et al., 2023) is another interesting review that analyses 16 CNN-based methods that establish the severity of damage over plants. This review also explains the whole methodology that must be tackled from the identification and understanding of the problem, the agreement in the annotation process, the selection of the network and design of the solution, and the evaluation itself. Their methods evaluate the damage on the leaves of the plants. A step further is done by (Shoaib et al., 2022) that propose a method that calculates the degree of damage and later, estimates the survival of the plant using a hybrid method combining CNN and linear regression. Once again, the damage estimation is done over leaves and not over the whole plant, and in this case, public Plant Village Dataset is used which is a not realistic dataset in terms of leaves that appear as single objects over a uniform background. In the field, the leaves usually appear overlapped and in non-uniform illumination conditions with variable backgrounds.

The only identified method that tackles the estimation of damage over the whole plant at a time is (Gómez-Zamanillo et al., 2023). The authors propose a method to assess the damage of soybean and redroot amaranth plants in the greenhouse through biomass estimation and deep learning-based symptom classification with reasonable accuracy. However, the plants in this approach are under controlled conditions in the greenhouse, isolated and with a uniform and white background. In this situation, the segmentation of the plant is feasible since it is isolated and there does not exist any overlap with any other plant.

In summary, there are some methods that establish the damage on the leaf of the plant, but only one work has been identified that addresses the estimation of the damage on the whole plant in a greenhouse under controlled conditions and isolated plants. On-leaf methods work under ideal conditions, such as a single leaf on a uniform black background. There are hardly any applications that analyze either leaves or plants outdoors, in the wild where lighting conditions and image quality are changing.

Our method must face a double problem. First, the detection of oilseed rape (BRSNN) plants in the image. The key issue is that other species might appear in the same image at a time. In fact, a multiple crop plant detection approach has been developed to make the model useful for other purposes in precision farming applications. Then, the degree of damage to each plant due to flea beetle is estimated.

3. Materials & methods

3.1. Dataset

An acquisition campaign took place in 2021 and 2022 in BASF in Spain, Canada, the USA, and Germany. Several crops were monitored. The selected crop to evaluate the method on was oilseed rape, whose EPPO code is BRSNN /BRSNS / BRSNW. Different trials were planned to have different situations for every crop and damage degree. Different damage was observed in the plants. The aim was to have a wide range of situations where the algorithm must detect oilseed rape plants and assess their damage. These images were acquired in the field under different natural illumination conditions (cloudy, sunny days) for several weeks. Images were taken with different mobiles and cameras. The field of view and distance to the plants are similar in all the images.

Two datasets have been prepared. One is dedicated to the development of crop plant detection, that is, the object detection model. The second one is available for the regression model for the estimation of damage in single plants. The images are totally different in these cases. The images for the object detection model cover a large area of the plot. Images for the regression model contain only single plants. An example of both is shown in the next image (See Fig. 5).

The dataset for object detection is composed of 5 target class crops. The detection model includes a total of 1125 images of 5 crops, of which 298 correspond to BRSNN trials. From there, 10% of the data samples were excluded randomly from each data trial to compute the test metric.

For the damage estimation model, there were two versions of the dataset. The first version was obtained from images of plots from trials in March 2022, and the second version was gathered in September 2022 of abovementioned Table 1. The plants images are extracted from the manual annotation of the images of plots. The same training, validation, and test distribution is guaranteed in the development of

Table 1
Summary table of the origin, year and number of images per crop for the object detection model.

CROP	EPPO CODE	Site	Year	N° images
Oil seed rape	BRSNN	Canada	2021	117
Oil seed rape	BRSNN	Spain	2022	85
Oil seed rape	BRSNN	USA	2022	96
Soybean	GLXMA	Spain	2020	5
Soybean	GLXMA	Spain	2021	118
Cotton	GOSHI	Spain	2020	198
Sunflower	HELAN	Spain	2019	70
Sunflower	HELAN	Spain	2021	210
Corn	ZEAMX	Germany	2020	8
Corn	ZEAMX	Germany	2021	3
Corn	ZEAMX	Spain	2021	125
Corn	ZEAMX	USA	2021	3
Corn	ZEAMX	IN	2021	9
Corn	ZEAMX	Spain	2022	3

both models. In the case of the dataset for the damage estimation model, it is important to analyze the instance distribution of images of plants and their related damage. The distribution of the images in the dataset with respect to damage and the number of images is as follows (See Fig. 6).

The total number of images in the first version of the dataset was 2243. The final number of images in the second version of the dataset is 2640. An irregular distribution can be observed. This is a problem known as class imbalance that may affect the performance of the developed models. There are some techniques that can address this problem, but they are not always effective. Moreover, there are some ranges of damage without any image. In this way, it is impossible for the model to learn. The first agreed decision was to limit the degree of damage that the model should predict. Therefore, the first version of the damage estimation model was designed and trained to estimate damage up to 40%. The second version of the model was designed and trained to estimate damage up to 60%.

3.2. Annotation process

All images in the dataset were annotated by experienced technicians. The annotation process consists of marking a bounding box around each crop plant (BRSNN, GLXMA; HELAN, ZEAMX, & GOSHI). For the BRSNN crop, an additional label was added to indicate the degree of damage that the expert found in the plant.

CVAT has been the annotation tool used to generate the annotations. Fig. 7 shows the CVAT user interface and an image that is being annotated.

The process of assigning a damage value to every plant can present difficulties due to the resolution of the images and depending on the fact that the plants appear isolated or in bunches. If they appear overlapped and part of the plant is not clearly visible, it is difficult for the expert to determine the degree of final damage with certainty. Evidently, this will be a drawback for the deep learning-based model, which can only infer over the visible pixels of the images and not over the hidden parts.

4. Proposed solution

The complete solution is a two-stage algorithm. The first stage aims to develop a deep learning-based detection algorithm based on the YOLOv5 architecture to detect single plants. The second stage of the algorithm takes all single detected plants only of oilseed rape crop (BRSNN) and an ad-hoc regression model estimates the damage degree for every plant.

4.1. Multicrop Plant detection

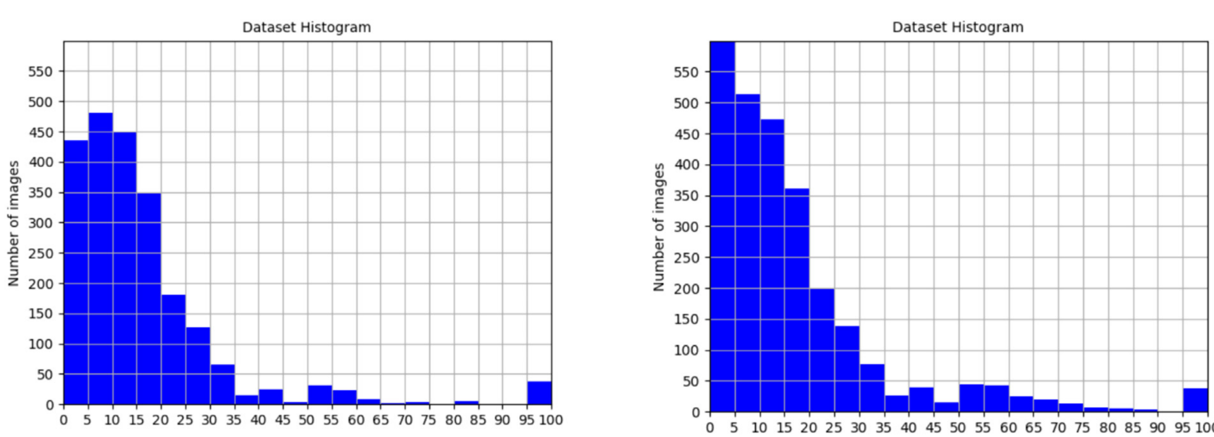
Detection of crop items is generally difficult. They look similar in colour, texture, and shape. However, deep learning has recently been shown to be a powerful technique for image understanding topics. This work will validate this assumption.

Creating a model to detect objects is a process of collecting and organizing images, labeling the objects of interest, training a model, deploying it, and optionally using that deployed model to collect further examples and improve the detection.

The original images are too large to be processed directly by the neural network, so an essential preprocessing step is image tiling or patching. Each plot image is partitioned into smaller tiles to standardize input image size to the network without losing resolution, and therefore improving detection accuracy. Furthermore, an overlapping stride window approach was followed, leading to 4 times more training samples. The overview of the complete workflow is summarized in the following figure.

After a few experiments, we decided to train a model with a tile size of 2048×2048 and an overlapping stride of 1024 in both width and height directions. This further avoids the main limitation of cutting objects. It ensures that the object is at least represented in the center of the tile once. To generate a fixed number of tiles, we decided to pad the image borders to a multiple of 1024 pixels. An illustration of the detailed process can be understood in Fig. 8. As we already mentioned, after preprocessing the number of instances per class was substantially increased. Additionally, to the tiling operation, the labels are converted to local coordinates in pixels and stored according to the Yolov5 specific format. Yolov5 expects the normalized labels in a .txt file containing class labels (zero-indexed) formatted as [class index, x centre, y centre, width, height]. (See Fig. 9.)

YOLO-based detectors are considered the state-of-the-art detection algorithms due to high accuracy and inference speed (Redmon et al., 2015). Our baseline detection model is Yolov5 architecture (Ultralytics, 2022), which is composed by a backbone (CSPDarknet), a neck (PANet), and head layer (yolo detection) as head layer for a single-stage detector. The network architecture is shown Fig. 10.



a b

Fig. 6. Distribution of number of images according to damage. A) distribution of the number of plant images and damage in the first version of the dataset; (b) final distribution of the number of plant images and damage in the final version.

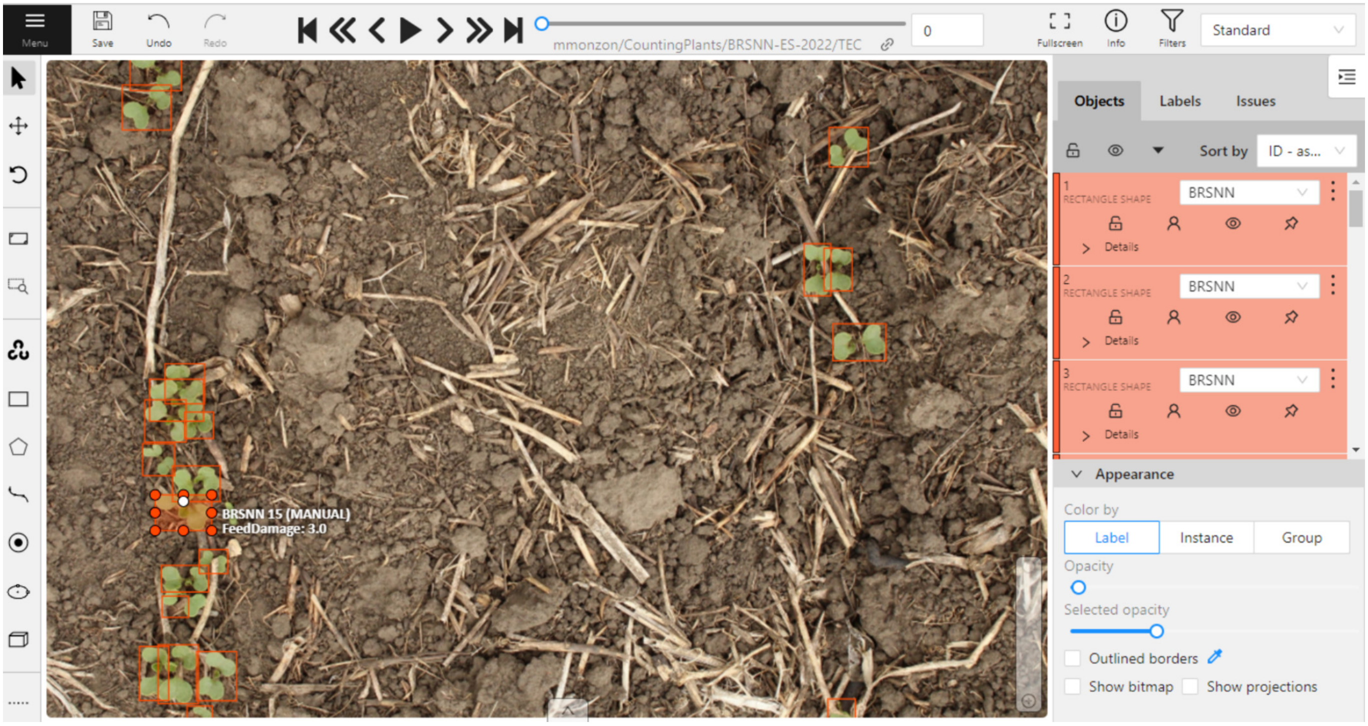


Fig. 7. Example of the annotation process on the CVAT Tool. Each target crop plant is annotated with a bounding box and the damage is indicated as each plant attribute.

Specifically, Yolov5 has the main advantage in dealing with images crowded with objects. It splits the input images into SxS cells, and then in every cell N number of bounding boxes are predicted.

The training of the object detection network is performed based on a supervised learning technique that requires input-target bounding box pairs. In supervised learning, a model is built by examining many examples and attempting to find a model that minimizes the loss function. In our case, the loss function is composed of three main penalization terms (Ultralytics, 2022):

- *objectness (obj):* term penalizing the finding of bounding-box coordinates finding. It quantifies the probability that an object exists in a proposed region of interest.
- *bounding-box regression loss (bbox):* the complete Intersection over Union (CIoU) loss is implemented, which incorporates normalized central point distance and aspect ratio factors to intersection over union metric.
- *classification score (cls):* penalizes the training based on the so-called binary cross-entropy with logit loss.

Yolov5 training process includes strong data augmentation techniques such as mosaic, flip, scale, translate, HSV colour space variation, … The parameters are stored for the Yolov5 hyper-parameter configurations along with the learning rate, the optimizer momentum rate and the data augmentation probabilities. These hyperparameters were evolved for the presented use case based on a Genetic algorithm (Jocher et al., 2022). After experimenting, the yolov5m6 variant of the model was trained for 200 epochs from the pre-trained model on COCO with AdamW (Loshchilov and Hutter, 2019) with the evolved hyperparameters for 100 evolutions. The main hyperparameters of the process are listed in Table 2.)

4.2. Damage estimation regression model

Once individual plants are detected, it is necessary to estimate the damage of each plant. To do so, a deep learning-based regression model has been developed and trained. The model has been trained on a supervised basis, where pairs of input images and output damage

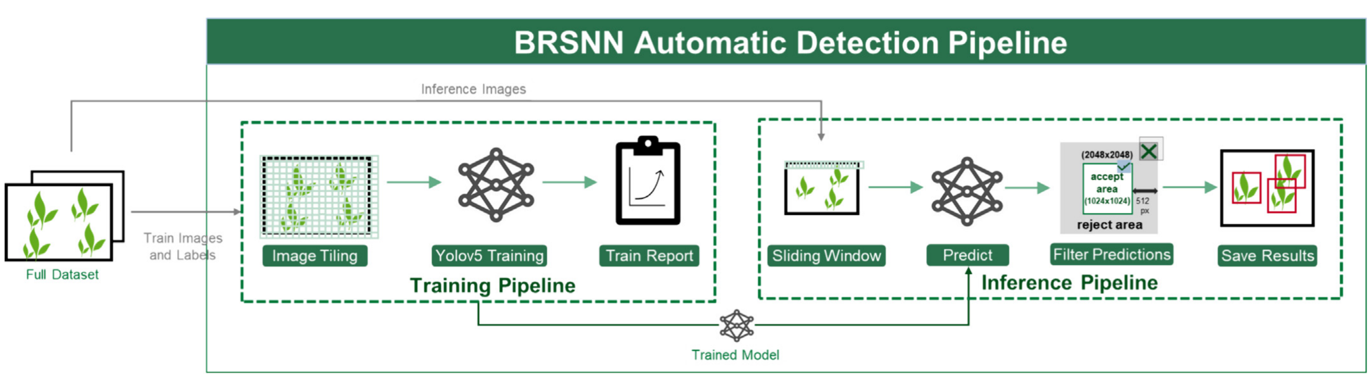


Fig. 8. Overview of the complete detection pipeline. The training pipeline is composed of a preprocessing of tiling the image, tuning the model weights, and creating a report showing the training learning curves and validation metrics. In the inference or test step, the images are tiled following the analogous methodology as in training, getting the prediction.

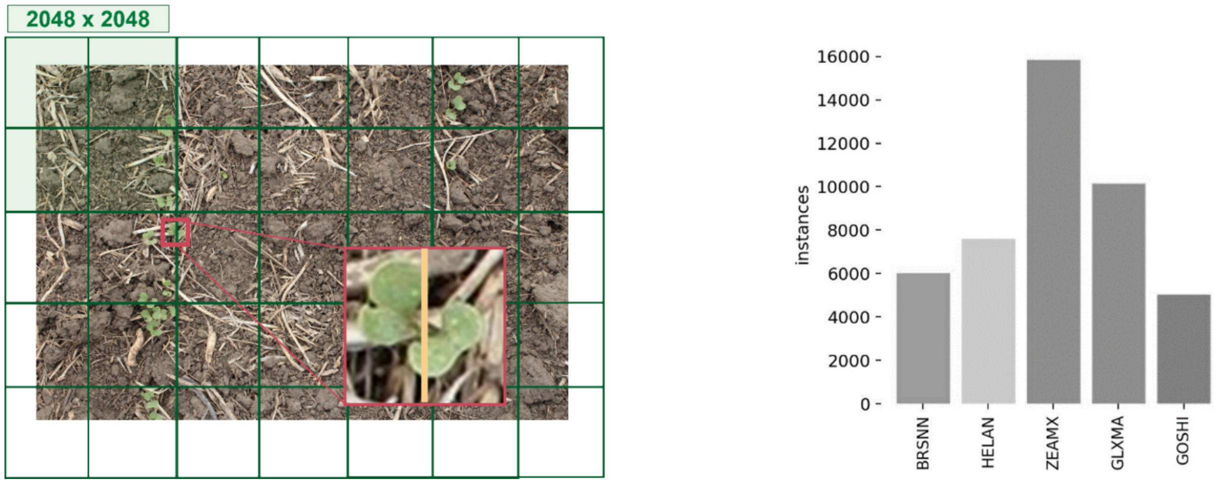


Fig. 9. Left. The image is tilted into smaller patches of 2048 × 2048 is shown. An overlapping stride window of 1024 × 1024 is applied after padding the image to get a fixed tile size for every stride. Right, the total number of instances of objects which were the input for the training step is depicted by class.

values are provided to the network to make it learn the features in the image that are relevant for the damage estimation.

Two different models have been developed. The first model was developed with the first version of the dataset. Due to the scarcity of data in certain ranges, it was decided that the model would estimate only damage up to 40%. The second model was developed with the second version of the dataset, and it estimates damage up to 60%.

To do so, the dataset was split up in training, validation, and test. In this split, the unbalanced distribution of the images according to the damage degree has been considered to make sure that there are examples (if available) of all the situations of damage in the three datasets: training, validation, and testing. The final figure of these groups of images is gathered in the Fig. 11.

For the second version of the model to be developed with dataset v2, it is aimed to develop a model capable of estimating damage degree up to 60%. Due to the high unbalanced dataset, mainly for plants with damage in the range 30–60%, it was decided to perform oversampling. A number of 400 samples per damage degree was decided to be applied. Augmentation techniques were used to perform this oversampling. These techniques contain affine transformations, such as flip, rotate, shift; intensity changes such as brightness, contrast and gamma modifications; and finally changes in colour appearance in two colour spaces, RGB values shift and HSV values shift. Fig. 12 shows the initial distribution of the dataset v2 and the final distribution of the images according to the damage after applying the oversampling techniques. (See Fig. 13.)

Several network topologies were tested as backbones, such as ResNet-based topologies with a different number of layers (ResNet18, ResNet34, ResNet50, ResNet101), efficientNet and efficientNetv2 families. Once the network architecture of the models was designed, it was necessary to start the trainings. From the experiments, it was concluded that eficientNetv2L was the most suitable for our problem. Dropout was added, but it did not improve the results. Different optimizers (SGD, RMSProp, Adam) and SGD was the best performing one. Learning rate was also modified, being 0.001 its final value. Different input sizes were tested, being 224 × 224 pixels size the one achieving better metrics. The weight initialization from ImageNet was also used. Additional experiments were performed with respect to the aspect ratio of the input image. Preserving the aspect ratio of the input image did not increase the results.

In the same way, an algorithm was designed to remove the background of the image. Therefore, only the plant would be seen by the network and not the background. This idea comes from the observation of the images where different backgrounds are possible, containing most of them sticks, dead leaves, and other elements that might be confused with damaged regions. However, removal of the background did not

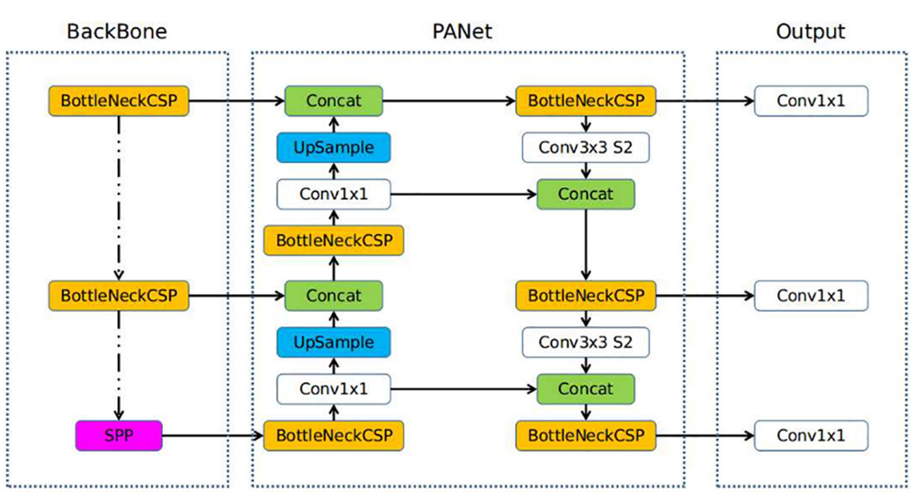


Fig. 10. Yolov5 network architecture. Image from (Jocher, 2022).

Table 2
Summary table of the training hyperparameters for the object detection model.

Hyperparameter	Value
Initial learning rate (lr0)	0.00072
final learning rate (OneCycleLR lrf)	0.08333
Adam optimizer momentum	0.84188
box loss gain coefficient	0.06218
Momentum	0.84188
Iou training threshold	0.1
Anchors per output layer	2.77461

improve the performance of the model. The next figure shows the output of the algorithm for background removal.

Another idea was tried by using the images of the plants without background. Instead of using only original images of the plant or with removal of the background, both types of images were included in the dataset as a data augmentation technique. This action contributed to the increase in the metrics of the model.

Data augmentation techniques were applied during the training process to increase the variability of the images seen by the network. Only affine transformations were applied in the training of the first model. Modifications in brightness, contrast and gamma and shift of the values in channels RGB and HSV of those colour spaces were included in the second version of the model. Test Time Augmentation (TTA) with 10 images was tested to assess whether the results improved. The final output is the average value of the 10 inferences on the augmented images of the original input image. The results are more accurate. Therefore, TTA with 10 images and with the augmentation operations included in the training stage was included.

5. Results

The described algorithms were developed on Python programming language and deployed as a service on a Linux based processing server. The complete system is provided as a docker image. These algorithms are based on Deep learning paradigm using Tensorflow and Pytorch frameworks as backend. The deployed service was prepared with REST Application Programming Interface that managed the connections from smartphone applications. The processing time of the algorithm was 2–4 s depending on the resolution of the input images, the higher time obtained for the size of the 4000 × 6000 pixels images, which corresponds to the higher resolution of the camera. Response time is acceptable, even if it could be improved in the future. To make this response time feasible, the plant detection model is executed in GPU. The execution of that model in CPU takes around 40 s; meanwhile, the execution on GPU takes no longer than 1–2 s. The execution of the regression model for damage estimation over 224 × 224 pixels size single plants is very similar in GPU or CPU.

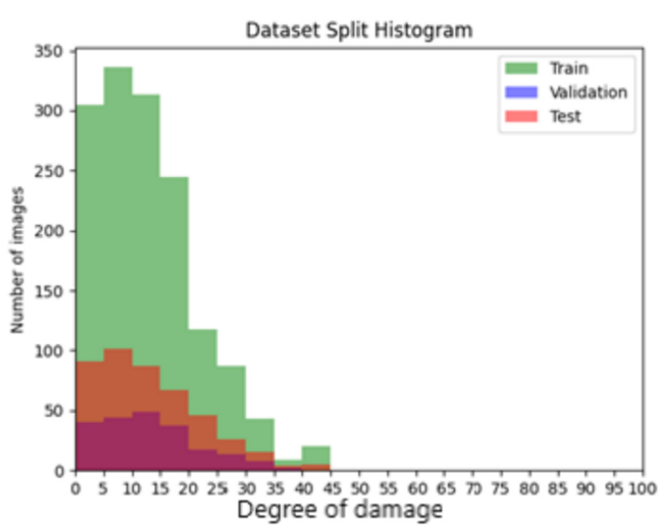


Fig. 11. Split up of the dataset v1 (left) into train, validation and test.

An example of a final output image and predicted values of damage is shown in the following Fig. 14:

Several tests were performed to validate the results of the proposed approaches for the different cases. Metrics are to be established to make comparison possible, to know the performance of the different models and decide which of them performs better. The results of all the models developed in this work are shown below. Comparison and discussion are also included.

5.1. Results for multicrop plant detection model

In object detection problems, various metrics are used to report the accuracy and performance of an object detection model. As detection is a joint problem of object recognition and classification, the most common evaluation method in classification can also be computed, the so-called confusion matrix (CM). The matrix represents the relation between correctly classified samples, i.e., True Positives (TP) and True Negatives (TN), and wrongly predicted samples, i.e., False Negatives (FN) and False Positives (FP) for each class. The CM can be displayed with absolute frequency values or normalized by each class total number of elements and ranged from 0 to 1 (this represents the best possible score). The confusion matrix is usually represented for a binary classification, although it can be extended to multiclass problem.

From the confusion matrix to assess the quality of the predictions further evaluation metrics could be derived such as precision, recall, F1-score, mean average Precision (mAP) at various levels (Padilla et al., 2020). These metrics can be further combined to form a new average metric to judge the performance (see Table 3). Fig. 15 shows the confusion matrix and the F1 score curve.

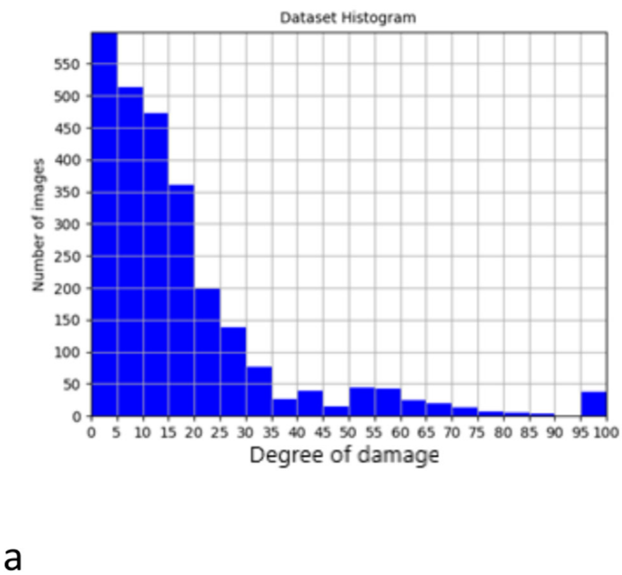
The most common metric in object detection problems is Mean Average Precision (mAP). It represents the average precision (AP) value for the recall value over 0 to 1. The average precision (AP) represents the area under the curve (AUC) Precision Recall Curve. AP is a numeric metric to summarize the precision-recall curve into a single value representing the average of all precisions. The mAP is an average of the AP values, which is a further average of the APs for all classes. The summary metrics for the model are represented here.

5.2. Results for damage estimation model

Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) have been established as metrics, together with R^2 for the regression model for damage estimation. This R^2 value is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination. A desirable value of R^2 is 1.0. It means that there is no error in the regression and that the predicted values fit a perfect line with a slope of value 1.0 in relation to the ground truth values. An R^2 of value 0 means that the predicted values are not better than taking the mean value of the x-axis values.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) present similarities with the human understanding, since they represent the direct relation between the predicted value and the real value. The metrics are calculated as follows:

$$MAE = \frac{\sum_{i=1}^{N} \left(y_i^{EST} - y_i^{GT}\right)}{N}$$



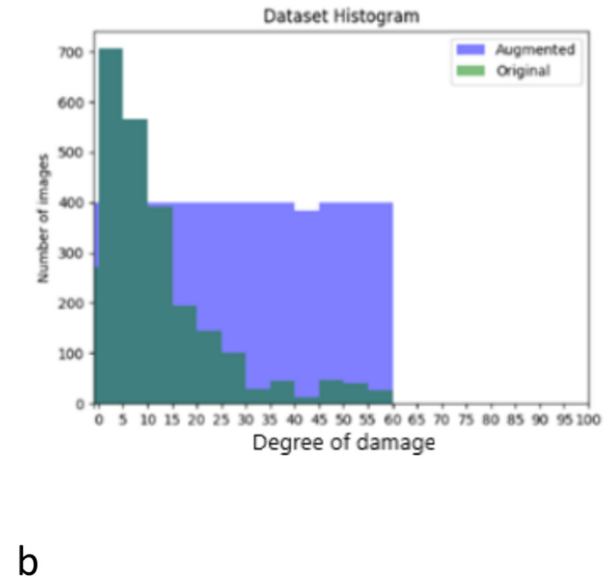


Fig. 12. a) Dataset v2; b) final oversampled dataset v2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i^{EST} - y_i^{GT}\right)^2}{N}}$$

In all cases, the metrics shown are the average values obtained for the testing dataset.

As it has been indicated in the previous section there have been two versions of the model. The first version has been developed with dataset v1 and can estimate damage up to 40%. The second version has been developed with dataset v2 and can estimate damage up to 60%. Results for both models are shown below.

The best results for model v1 have been obtained with 'efficientnetv2L' backbone and training with both types of images (with/without background). Validation and test were made over images with background. The metrics obtained are MAE of 5.44, RMSE of 6.97 and R2 of 0.36 (see Fig. 16).

Model v2 has been developed with 'efficientnetv2B2' as the backbone and training with both types of images (with/without background). Validation and testing were performed over images with background. The obtained metrics are MAE of 7.11, RMSE of 9.66 and R2 of 0.46 (See Fig. 17). The test dataset of the v2 model contains 224 images of plants. The selection of the model v2 has been made taking into consideration the performance metrics but also the inference time. Several topologies were compared in terms of both accuracy and response time. Backbones such as 'efficientnetv2L' are around 440 MB in size, much larger than the 39 MB of 'efficientnetv2B2'. The larger the model size, the longer the inference time. Moreover, the inclusion

Fig. 13. Examples of original images of plants and those plants without background.

Fig. 14. Example of output image with the detected plants and the estimated damage for any of the plants.

of Test Time Augmentation technique implies that 10 inferences are made per plant image, and there are many plants in every image of the plot. In the selection of the final model, together with the metrics, the training graphics and the histogram with the means absolute cumulative error have been considered. This is because the highest number of images with error are concentrated in a small error range. This fact has implied the final selection of 'efficientnetv2B2' as the best architecture.

The final regression plot and metrics are depicted in the following figure.

In Fig. 18 we can see that the final model achieves the following:

- 83% of the test images with absolute error less or equal than 10
- 91% of the test images with absolute error less or equal than 15

The test has been done on a subset of 224 plant images.

Additional analysis of the results provided by the model has been done by means of box-and-whisker plot, where maximum, minimum, mean, and median values are shown. Errors in the different ranges of damage can be appreciated. Comparison of the error in the different ranges can be done in relation to the final histogram of the dataset. In this way a real number of plant images and oversampled ones per damage range can be appreciated. This helps in the extraction of the conclusions.

6. Analysis of the results

Once the results of individual performance for both models over the test images have been shown, we expose here our analysis of the results.

The multi-crop plant detection algorithm using tiling and YOLOv5 architecture is feasible and robust for various degrees of damage to plants. It has been validated in the damage range 0–60%. The overlapping sliding window approach enables a high accuracy detection model, despite a higher computational requirement. The performance of the detection algorithm is robust to detect each individual crop plant to compute the individual disease estimation. The mean average precision of the proposed model for BRSNN, oilseed rape, our target crop for validation, is 0.99. Currently, the multicrop plant detection is capable of detecting plants of 5 different crops, that is, BRSNN, ZEAMX,

Table 3
Summary table of the number of instances per crop, and the test metrics: precision, recall, mean average precision at IoU 0.5 and mean average precision at IoU 0.95 for the object detection train model training.

Class	Labels	P	R	mAP@0.5	mAP@0.5:0.95
BRSNN (Oilseed rape)	640	0.971	0.990	0.992	0.909
HELAN (sunflower)	913	0.934	0.908	0.963	0.752
ZEAMX (corn)	1308	0.901	0.836	0.888	0.539
GLXMA (soybean)	1043	0.867	0.827	0.859	0.616
GOSHI (cotton)	587	0.977	0.952	0.983	0.897
ALL	**4491**	**0.930**	**0.903**	**0.937**	**0.743**

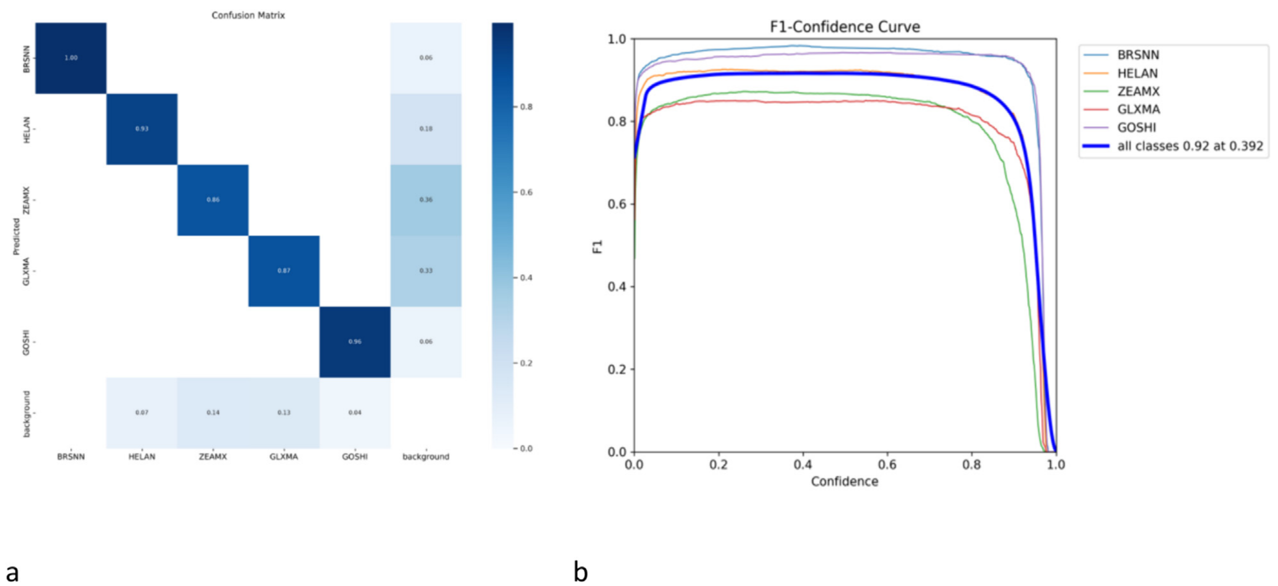


Fig. 15. Test performance metrics results. (a) confusion matrix for the test dataset. (b) F1-confidence curve.

HELAN, GLXMA and GOSHI with high accuracy for them all, BRSNN being the one showing higher metrics.

However, some drawbacks have been identified during the testing phase. Together with oilseed rape, BRSNS crop, the malva weed usually appear. Sometimes 1MALG (malva) plants are detected as oilseed rape. Even if the confidence level is checked and those plants with confidence level lower than 0.7 are discarded, some of them are still present. Therefore, these plants are passed through the damage estimation model, and the damage value is retrieved. It can be considered a false positive. A further version of the multicrop plant detection model can solve this issue by including malva in the crops/weeds it detects. An example of this situation is shown next.

Regarding regression model for damage estimation over single plants previously detected by the multicrop plant detection model, it can be said that performance of the model is quite good. There usually exist discrepancies between damage assessment done by experts and by algorithms, and experts tend to overestimate the damage. The current version of the model can detect damage up to 60%. However, the number of images in the dataset containing plants with damage greater than 30% was really small. Oversampling techniques were applied to try to balance the dataset, but in fact these added images are augmented images with few images as original source. This makes that the error in the prediction tend to increase in plants with damage higher than 30%. This can be appreciated in Fig. 19. This is referred to in the literature and seems to be logical since the oversampled images are augmented images over few original ones, and do not provide enough variability to the model. Therefore, the generalizability of the model is not good enough. Techniques applied to minimize the unbalanced dataset can never replace the contribution of real images.

The regression model has provided an R^2 of 0.46 (slope of 0.54) and MAE of 7.11 on 224 plant images in the test subset. From the analysis of the error histograms, it has been seen that 83% of the test images have

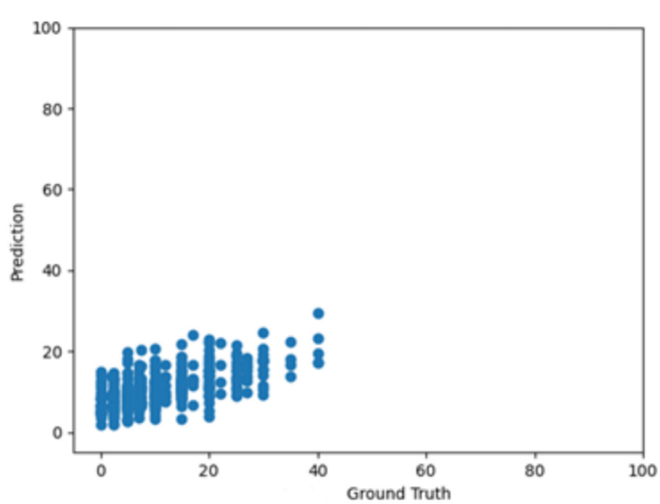


Fig. 16. Results of damage estimation regression model for dataset v1 and damage range up to 40%.

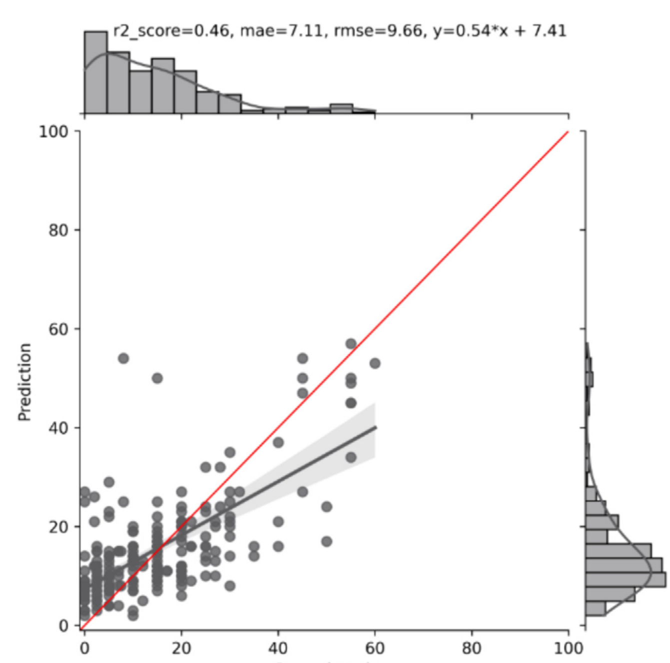
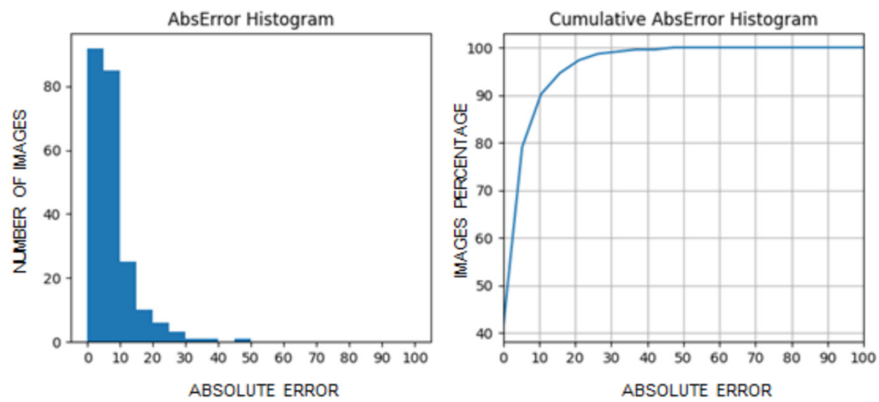


Fig. 17. Results of damage estimation regression model for dataset v2 and damage range up to 60%.



label	count	perc
[0-5]	110	49%
(5-10]	77	83%
(10-15]	16	91%
(15-20]	9	95%
(20-25]	7	98%
(25-30]	2	99%
(30-35]	2	100%
(35-40]	0	100%
(40-45]	0	100%
(45-50]	1	100%
(50-55]	0	100%
(55-60]	0	100%

a

b

Fig. 18. a) Cumulative absolute error for the test images; b) detailed number of images per absolute error range.

an absolute error less than or equal to 10 and 91% of the test images have an absolute error lower than 15. The discrepancy (absolute error) among the annotators has been estimated in the range 10–15, and it can also vary in different moments of time for the same expert. This is known as inter-rater and intra-rater variability. An experiment was carried out during the project with a set of images that presented a high error with the initial annotations. A second annotation process was carried out by the same experts, and the variations between the two annotations values of the same set of images were of 14.66% in the value of the damage.

Therefore, the MAE provided by the model is lower than the error that experts might provide. Slope of 0.54 indicates that the model provides lower values than the expert (ideally, the slope of the regression line should be 1). R^2 of 0.46 also shows this dispersion. This dispersion revealed by R^2 is also affected by errors due to overlapping plants. The plants sometimes appear in bunches and their leaves overlap. This implies that the damage estimation model does not see all the leaves of the plant to be analysed and sees other leaves (see Fig. 21). This might not happen in the field, as the expert can appreciate all the leaves in the plant directly and can prevent occlusions. Fig. 20 shows this limitation of the method. This limitation affects the training process as well; noisy annotations are included in the training process. This could be partially solved with an instance segmentation solution instead of the object detection approach for the multicrop plant detection model. This would avoid the inclusion of leaves of other plants; however, this would not solve the hidden information of the occluded leaves. Anyway, an instance segmentation approach for the crop detection seems challenging, especially in the bunches where it is often

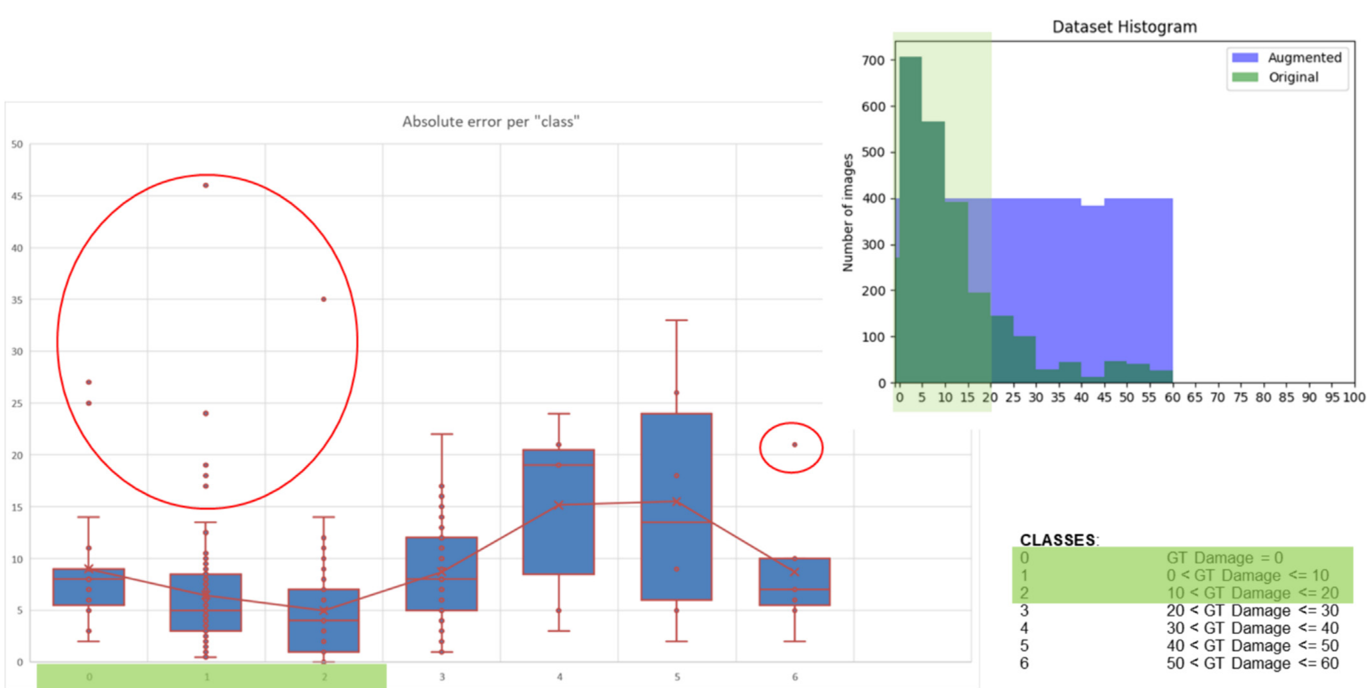


Fig. 19. Box and whisker plot that represents error dispersion in different damage ranges. Dataset histogram with the distribution of plant images according to damage values.

Fig. 20. Detection of malva plant by the multicrop plant detection model, and further obtained damage estimation.

difficult for a human to establish the contour limits among plants over images for its manual annotation.

Another detected issue is that the model for damage estimation finds it difficult to estimate 0% damage, predicting always values lower than 5% in healthy plants. Probably waterdrops, shines, or other elements provoke this effect. Ad-hoc solution should be found to address this limitation and improve the results.

7. Conclusions and next steps

In this work a two-stage method has been presented for the estimation of damage in plants in plots in the field. The obtained results are considered good for both developed models. The current version of the algorithm can detect and estimate damage in plants of up to 60%.

Further work can be focused on including new images in the dataset not available now to develop a new model to estimate damage up to 100%. To achieve that, plants with a damage degree higher than 60% should be included in the dataset. Furthermore, to reduce the error in the damage range from 40 to 60%, more plants with this damage should be added. Another way of improving the results could be to include plots with malva or other potential weeds that may appear in new versions of multicrop plant detection models. This way, the number of false positives due to this weed that can be confused with oilseed rape will be reduced. In addition to this, additional strategies must be added to deal with highly damaged plants. Plants with high damage are completely eaten by flea beetles and are missing plants in the plot. Therefore, they will not be detected by the multicrop detection model in the first stage. This is a real drawback that must be considered mainly for plants with damage higher than 85 or 90%. Maybe tracking on plant locations in the field is to be done to include the expected position of the plants in the location algorithm for its analysis.

Fig. 21. Example of detected plants with overlapped leaves of adjacent plants.

Another research line to work on is to design a strategy to improve the identification of plants with 0% damage. It could be possible to build a new architecture with two branches: one for the classification of the plant as with damage or without damage, and the second branch for a regression model to estimate the damage in case damage is classified as higher than 0%.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwal, M., Gupta, S.K., Biswas, K.K., 2020. Development of efficient CNN model for tomato crop disease identification. Sustain. Comput. Informatics Syst. 28, 100407. https://doi.org/10.1016/j.suscom.2020.100407.

Ahad, M.T., Li, Y., Song, B., Bhuiyan, T., 2023. Comparison of CNN-based deep learning architectures for rice diseases classification. Artif. Intell. Agric. 9, 22–35. https://doi.org/10.1016/j.aiia.2023.07.001.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2019. Few-Shot Learning Approach for Plant Disease Classification over Field Images.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2020. Few-shot learning approach for plant disease classification using images taken in the field. Comput. Electron. Agric. 175. https://doi.org/10.1016/j.compag.2020.105542.

Bai, X., Liu, P., Cao, Z., Lu, H., Xiong, H., Yang, A., Cai, Z., Wang, J., Yao, J., 2022. Rice Plant counting, locating and sizing Method Base on high-throughput UAV RGB images. Plant Phenomics 1–16. https://doi.org/10.34133/plantphenomics.0020.

Bereciartua-Pérez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., Eggers, T., 2022a. Insect counting through deep learning-based density maps estimation. Comput. Electron. Agric. 197, 106933. https://doi.org/10.1016/j.compag.2022.106933.

Bereciartua-Pérez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., Eggers, T., 2022b. Multiclass insect counting through deep learning-based density maps estimation. Smart Agric. Technol. J. 3. https://doi.org/10.2139/ssrn.4184417.

David, E., Daubige, G., Joudelat, F., Burger, P., Comar, A., de Solan, B., Baret, F., 2021. Plant detection and counting from high-resolution RGB images acquired from UAVs: comparison between deep-learning and handcrafted methods with application to maize, sugar beet, and sunflower crops. bioRxiv 441631 2021.04.27.

Esgario, J.G.M., Krohling, R.A., Ventura, J.A., 2020. Deep learning for classification and severity estimation of coffee leaf biotic stress. Comput. Electron. Agric. 169. https://doi.org/10.1016/j.compag.2019.105162.

Fuentes, A., Yoon, S., Kim, S.C., Park, D.S., 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors (Switzerland) 17. https://doi.org/10.3390/s17092022.

Gao, G., Gao, J., Liu, Q., Wang, Q., Wang, Y., 2020. CNN-Based Density Estimation and Crowd Counting: A Survey. pp. 1–25.

Ghosal, S., Blystone, D., Singh, A.K., Ganapathysubramanian, B., Singh, A., Sarkar, S., 2018. An explainable deep machine vision framework for plant stress phenotyping. Proc. Natl. Acad. Sci. 201716999. https://doi.org/10.1073/pnas.1716999115.

Gómez-Zamanillo, L., Bereciartua-Pérez, A., Picón, A., Parra, L., Oldenbuerger, M., Navarra-Mestre, R., Klukas, C., Eggers, T., Echazarra, J., 2023. Damage assessment of soybean and redroot amaranth plants in greenhouse through biomass estimation and deep learning-based symptom classification. Smart Agric. Technol. 5. https://doi.org/10.1016/j.atech.2023.100243.

Hosseiny, B., Rastiveis, H., Homayouni, S., 2020. An automated framework for plant detection based on deep simulated learning from drone imagery. Remote Sens. 12, 1–21. https://doi.org/10.3390/rs12213521.

Houetohossou, S.C.A., Houndji, V.R., Hounmenou, C.G., Sikirou, R., Kakaï, R.L.G., 2023. Deep learning methods for biotic and abiotic stresses detection and classification in fruits and vegetables: state of the art and perspectives. Artif. Intell. Agric. 9, 46–60. https://doi.org/10.1016/j.aiia.2023.08.001.

Ji, M., Zhang, K., Wu, Q., Deng, Z., 2020. Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. Soft. Comput. 2. https://doi.org/10.1007/s00500-020-04866-z.

Jiang, Y., Li, C., 2020. Convolutional neural networks for image-based high-throughput plant phenotyping: a review. Plant Phenomics 2020, 1–22. https://doi.org/10.34133/2020/4152816.

Jocher, G., 2022. Ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime instance segmentation (v7.0). Https://Github.Com/Ultralytics/Yolov5/Tree/V7.0 10. https://doi.org/10.5281/ZENODO.7347926.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., 2022. Ultralytics/yolov5: v6.2 - YOLOv5 classification models, apple M1, reproducibility, ClearML and Deci.Ai integrations (v6.2). Zenodo. [WWW document]. https://doi.org/10.5281/zenodo.7002879.

Johannes, A., Picon, A., Alvarez-gila, A., Echazarra, J., Rodriguez-vaamonde, S., Díez, A., Ortiz-barredo, A., 2017. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput. Electron. Agric. 138, 200–209. https://doi.org/10.1016/j.compag.2017.04.013.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

Li, L., Zhang, S., Wang, B., 2021a. Plant disease detection and classification by deep learning - a review. IEEE Access 9, 56683–56698. https://doi.org/10.1109/ACCESS.2021.3069646.

Li, W., Zheng, T., Yang, Z., Li, M., Sun, C., Yang, X., 2021b. Classification and detection of insects from field images using deep learning for smart pest management: a systematic review. Ecol. Inform. 66, 101460. https://doi.org/10.1016/J.ECOINF.2021.101460.

Liu, W., Zhou, J., Wang, B., Costa, M., Kaeppler, S.M., Zhang, Z., 2022. IntegrateNet: a deep learning network for maize stand counting from UAV imagery by integrating density and local count maps. IEEE Geosci. Remote Sens. Lett. 19. https://doi.org/10.1109/LGRS.2022.3186544.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. 7th Int. Conf. Learn. Represent. ICLR 2019.

Padilla, R., Netto, S.L., Da Silva, E.A.B., 2020. A survey on performance metrics for object-detection algorithms. Int. Conf. Syst. Signals, Image Process, 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130 2020-July.

Picon, A., Alvarez-gila, A., Seitz, M., Ortiz-barredo, A., Echazarra, J., Johannes, A., 2019. Deep convolutional neural networks for mobile capture device-based crop disease classi fi cation in the wild. Comput. Electron. Agric. 161, 280–290. https://doi.org/10.1016/j.compag.2018.04.002.

Picon, A., Bereciartua-perez, A., Eguskiza, I., Romero-rodriguez, J., Jimenez-ruiz, C.J., Eggers, T., Klukas, C., Navarra-mestre, R., 2022. Arti fi cial intelligence in agriculture deep convolutional neural network for damaged vegetation segmentation from RGB images based on virtual NIR-channel estimation. Artif. Intell. Agric. 6, 199–210. https://doi.org/10.1016/j.aiia.2022.09.004.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2015. You only look once: unified. Real-Time Object Detection. https://doi.org/10.1109/CVPR.2016.91.

Sai Reddy, B., Neeraja, S., 2022. Plant leaf disease classification and damage detection system using deep learning models. Multimed. Tools Appl. 81, 24021–24040. https://doi.org/10.1007/s11042-022-12147-0.

Saleem, Muhammad, Hammad, Potgieter, Johan;Arif, K.M, 2019. Plant disease detection and classification by deep learning. Plants 8, 468. https://doi.org/10.3390/plants8110468.

Sharma, P., Berwal, Y.P.S., Ghai, W., 2020. Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Inf. Process. Agric. 7, 566–574. https://doi.org/10.1016/j.inpa.2019.11.001.

Shi, M., Li, X.Y., Lu, H., Cao, Z.G., 2022. Background-aware domain adaptation for plant counting. Front. Plant Sci. 13, 1–16. https://doi.org/10.3389/fpls.2022.731816.

Shi, T., Liu, Y., Zheng, X., Hu, K., Huang, Hao, Liu, H., Huang, Hongxu, 2023. Recent advances in plant disease severity assessment using convolutional neural networks. Sci. Rep. 13, 1–13. https://doi.org/10.1038/s41598-023-29230-7.

Shoaib, M., Shah, B., Hussain, T., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Ali, F., Syed, I., 2022. A deep learning-based model for plant lesion segmentation, subtype identification, and survival probability estimation. Front. Plant Sci. 13, 1–15. https://doi.org/10.3389/fpls.2022.1095547.

Singh, A., Ganapathysubramanian, B., Singh, A.K., Sarkar, S., 2016. Machine learning for high-throughput stress phenotyping in plants. Trends Plant Sci. 21, 110–124. https://doi.org/10.1016/j.tplants.2015.10.015.

Singh, A.K., Ganapathysubramanian, B., Sarkar, S., Singh, A., 2018. Deep learning for plant stress phenotyping: trends and future perspectives. Trends Plant Sci. 23, 883–898. https://doi.org/10.1016/j.tplants.2018.07.004.

Singh, V., Sharma, N., Singh, S., 2020. A review of imaging techniques for plant disease detection. Artif. Intell. Agric. 4, 229–242. https://doi.org/10.1016/j.aiia.2020.10.002.

Tian, H., Wang, T., Liu, Y., Qiao, X., Li, Y., 2020. Computer vision technology in agricultural automation —a review. Inf. Process. Agric. 7, 1–19. https://doi.org/10.1016/j.inpa.2019.09.006.

Ultralytics, 2022. YOLOv5: The Friendliest AI Architecture you'll Ever Use [WWW Document]. URL: https://ultralytics.com/yolov5.

Valente, J., Sari, B., Kooistra, L., Kramer, H., Mücher, S., 2020. Automated crop plant counting from very high-resolution aerial imagery. Precis. Agric. 21, 1366–1384. https://doi.org/10.1007/s11119-020-09725-3.

Zhang, Q., Liu, Y., Gong, C., Chen, Y., Yu, H., 2020. Applications of deep learning for dense scenes analysis in agriculture: a review. Sensors (Switzerland) 20, 1–33. https://doi.org/10.3390/s20051520.