Improving accuracy and efficiency in plant detection on a novel, benchmarking real-world dataset

2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor) | 978-1-6654-0533-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/MetroAgriFor52389.2021.9628717

Improving accuracy and efficiency in plant detection on a novel, benchmarking real-world dataset

1st Laurenz Ohnemüller
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
l.ohnemueller@student.maastrichtuniversity.nl

2nd Alexia Briassouli
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
alexia.briassouli@maastrichtuniversity.nl

***Abstract*—Detecting plants in images is central in precision agriculture, but can be challenging due to their small size, similarities in appearance, varying lighting and environmental conditions. Moreover, computational capacity in real-world settings may be limited. This work examines how accurate, computationally efficient real-time plant detection can be achieved on the large and varied benchmarking Open Plant Phenotyping Database, by building upon the State-of-the-Art (SoA) Scaled YOLO v4 real-time object detection model. The effect of pre-processing, namely cropping unnecessary information and increasing contrast, is examined and experimentally shown to improve both accuracy and efficiency. Transfer learning is also leveraged for the deployment of Scaled YOLO v4, using pre-trained weights from the MS COCO data set, and shown to lead to a moderate improvement in accuracy. The proposed final model results in approximately 10% higher accuracy than the existing baseline model, on a representative subset of about half of images in the Open Plant Phenotyping Database. Experiments show that plant detection accuracy is improved for most well represented samples, with errors appearing in particularly challenging cases or caused by data imbalance. This shows the proposed method has significant potential for highly accurate and computationally efficient plant detection in real-world environments.**

***Index Terms*—precision farming, object detection, transfer learning, YOLO, plant classification**

I. Introduction

Computer vision in agriculture provides a non-contact and non-destructive method of plant monitoring, similar to the way farmers visually inspect their crops. However, analyzing the resulting images can be complex and challenging [1], as growing cycles of crops and plants vary in size, height or shape [2]. If growth conditions, such as the soil and the availability of water or sun are changing, a different appearance of the plant can be expected [3]. Moreover, image acquisition and illumination can affect the appearance of the plant, while occlusion and often missing distinct edges can complicate the partitioning of individual plants [4].

Artificial intelligence, and specifically SoA object detection methods can be deployed to address these issues. These approaches heavily rely on large quantities of labeled training data, so large and especially diverse data sets are required to make models more generalizable and accurate. A variety of different data sets already exist [5], but most are either too small, too artificial or too homogenous. Recently, a large and varied dataset was made publicly lavailable, the Open Plant Phenotyping Database [6]. We propose a method for accurate, real-time plant detection on it, and compare it with the more costly, baseline model proposed by its authors [6]. Our approach comprises of a series of appropriate pre-processing steps, and the implementation of a SoA real-time and accurate object detection model, Scaled YOLO v4 [7]. We also examine the effect of transfer learning by deploying Scaled YOLO v4 pre-trained on MSCOCO [8], a different, but large and diverse dataset. Our results show that the pre-processing steps proposed, and deployment of pre-trained Scaled YOLO v4 lead to accurate real-time plant detection on a representative subset of this challenging dataset.

This work is organized as follows: related work is presented in Sec. II, and the dataset used in Sec. III. Details of the method used are presented in Sec. IV, while experimental results are presented in Sec. V, where the effect of pre-processing and transfer learning are also discussed. Conclusions and directions for future work are drawn in Sec. VI.

II. Related Work

The topic of intelligent systems in agriculture was first discussed by McKinnion and Lemmon (1985), who implemented an expert decision system for crop management [9]. The introduction of computer vision in this sector has built a very strong momentum in the last decade, with object detection and classification among the commonly addressed problems.

Today's SoA methods for object detection are based on deep learning, which uses convolutional neural networks (CNNs). CNNs take two-dimensional image data as input and pass it through multiple layers of neurons with corresponding weights, which perform operations like convolutions, pooling and others on the input to extract features. Through learning, the weights on each level can self-optimize to achieve more accurate predictions. Region-based CNNs (R-CNNs) [10] are one of the SoA object detection methods, which achieve very high detection accuracy by applying CNNs to different regions of an image. However, their computational requirements are high, as they apply CNNs to each region separately to classify it. Real-time, accurate object detection can be achieved by single-shot frameworks, such as the YOLO-framework (You Only Look Once) [11], that has shown great performance. YOLO is based on regression, and instead of selecting features

978-1-6654-0533-1/21/$31.00 ©2021 IEEE

of an image, it predicts classes and bounding boxes. By applying a CNN to the entire image, it significantly reduces the computational time of R-CNN.

In object detection for precision farming, Bargoti and Underwood [12] implemented a Faster R-CNN for accurate detection of three kinds of fruits in orchards. Gao et al. implemented tiny YOLOv3, to detect weeds in sugar beet fields [13]. Modified versions of YOLOv3 have achieved SoA detection of tomatoes in a real environment [14]. The Open Plant Phenotyping Database [6] contains different categories of plants at different growing stages, under varying growing conditions, with high intra-class similarity and occlusions from leaves. It has only been analyzed by its authors [6], who carried out object localization using a Faster R-CNN with ResNet50v1 as a feature extractor with pre-trained weights from the MS COCO data set [8]. All instances were changed to the same class, making it a pure object localization task.

In this work, we analyze the Open Plant Phenotyping Database using one of the latest versions of YOLO, namely Scaled YOLOv4 [7], as it has been shown to achieve high accuracy at a low computational cost and in real time, all important factors in plant detection for precision farming. In our experiments, we show that for a balanced subset of the dataset, we indeed achieve higher accuracy on the more challenging task of real-time plant detection than [6].

III. Data set

The Open Plant Phenotyping Database is the largest public data set of its kind, consisting of 7590 RGB images with 47 different plants. It features high inter-class similarity and high intra-class variability, as plants are captured over 4 trial seasons and in 3 different growth conditions: natural, drought, ideal. Each plant is in a polystyrene box, which we crop from each image during pre-processing.

IV. Methodology

In this work, the SoA object detection model Scaled YOLO v4 [7] is used, as it has been shown to exceed prior benchmarks from previous models. The best EfficientDet model [15] is slightly worse than the best Scaled YOLO v4, and requires more than twice the training time. While most detection systems are two-shot detectors that first predict the regions of interest and then classify it like the Faster R-CNN, YOLO is a single-shot detector. This means object localization and classification is done in a single pass of the network, which often makes the models smaller and faster. The general idea of YOLO is that a given image is split into a grid, where each cell predicts bounding boxes, confidences and class probabilities. Each of these predictions is done using a single CNN.

Scaled YOLO v4 stands out especially compared to prior YOLO-models, as it introduces an innovative technique of Cross-Stage Partial Networks [16]. This way the model can reduce an immense amount of computation. Cross-Stage Partial Networks work by splitting the output signal, where one half goes along the main path to generate more semantic information with a large receptive field, while the other half follows another branch obtaining more spatial information with a small receptive field. Next to that, it also uses the Mish Activation function [17], which outperforms all standard functions. As for the loss functions, which are used to calculate the error of the prediction and the ground truth, mainly three different kinds are used. A novel Complete Intersection over Union (CIoU) loss [18] describes the localization accuracy of the bounding box and overcomes previous issues of slow convergence and inaccurate performance. A classification loss is implemented for the conditional class probabilities and lastly an objectness loss that represents a confidence score for each bounding box. In other words, while the objectness loss tries to teach the model to make a correct object-box prediction, the CIoU loss aims to better predict the bounding box. Consequently, this results in all scaled YOLOv4 models being Pareto optimal. Because of limited computational power, the medium sized model, the YOLOv4-CSP, was implemented with all hyperparameters already initialized. However, it is possible to scale up for more accuracy by using one of the YOLOv4-P5/P6/P7 models. The model uses pre-trained weights from the MS COCO data set, as its size and diversity are expected to lead to robust object detection.

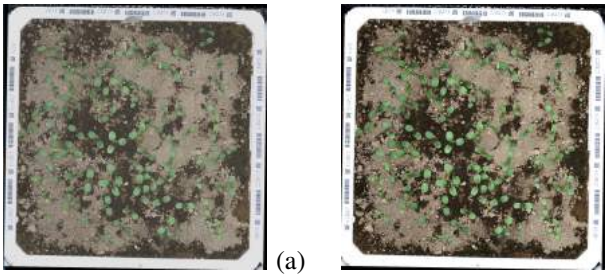

Fig. 1: Pre-processing with contrast enhancement: (a) Original image after cropping the black background (b) Image with 40% more contrast than the original and cropped background.

V. Experiments

In this section we present our experiments, where we demonstrate the influence of pre-processing, transfer learning and the resulting improved plant detection accuracy. The setup and the training process of the Scaled YOLO v4 is based on the Python version 3.6.9 with a NVIDIA Tesla T4 GPU. Besides, PyTorch framework version 1.8.1 was used in combination with CUDA 11.2 to utilize the GPU. PyYAML 5.4.1 was installed to read YAML files. The training process was the same for all experiments. The model trains for 100 epochs with a batch size of 16. The input images are of size 480×480 pixels and the hyperparameters' initialization stayed constant.

Dataset: A representative subset, comprising of approximately half of the Open Plant Phenotyping Database, was used, due to its high storage and computational requirements. This corresponds to 24 plant classes out of 47 and 3782 images in total. For each plant class, all available images are used, except those that were removed because of errors. The data set contains 9.22% of negative samples, which are

images without any plants. The data is not well balanced, partly because of the nature of the germination process, as some plants need more space, while other remain small. This could lead to poor predictive performance regarding minority classes. Transfer learning is also examined below, as it can provide a wider range of training data, potentially increasing the method's robustness. The data set is randomly split into a training, validation and testing set with a random seed at 30k, where each set has a similar percentage of negative samples, so that the results are not biased. The training set consists of 2731 images (72% of the entire data set), the validation set of 483 images (13% of the data), and the testing set contains 568 images (15% of the data). This split is done to avoid overfitting, where the model would learn features specific only to the training data.

Pre-processing: Cropping, contrast enhancement: Fig. 2(a) shows that cropping images leads to 7% higher mAP, which can be attributed to the removal of redundant background areas, that can lead to false alarms or misidentifications. Occlusions and irrelevant objects are also sometimes present in the area surrounding each plant box, making its detection more challenging. Fig. 2(b) shows that adding contrast improves the mAP by 2% as well. This is a minor improvement, but easy to achieve with this simple pre-processing step.

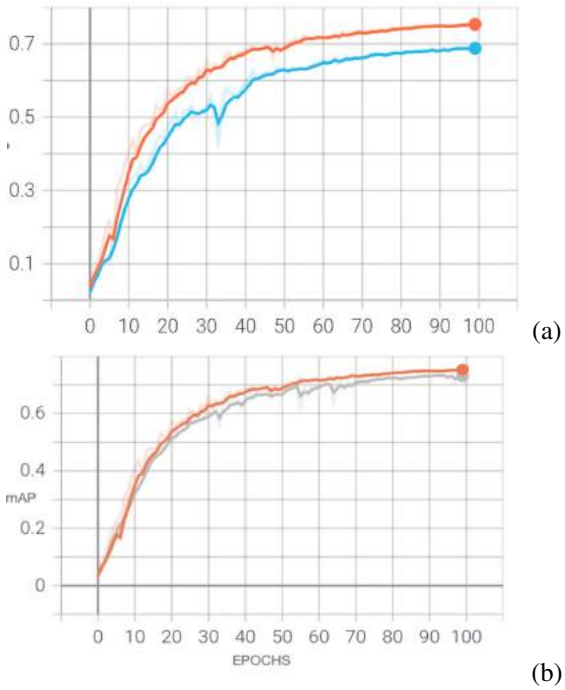

Fig. 2: mean Average Precision for Intersection over Union 0.5 $mAP@0.5$ during model training (best viewed in color): (a) with cropped images (orange graph), without cropped images (blue graph). (b) With contrast (orange graph), and without contrast (grey graph).

Transfer Learning: To test the influence of transfer learning, the model is trained under the same configuration setup on the same data, once with pre-trained weights from the MS COCO data set and once without. Fig. 2 (c) shows the advantage of pre-trained weights, as the model with pre-trained weights has a much faster training time, with a stronger increase in the first 40 epochs. Afterwards, both cases develop at a similar slower rate, until the pre-trained model reaches a roughly 2% higher mAP. This small improvement can be attributed to the fact that the MS COCO data set is not specified for plants, but rather for all kinds of objects.

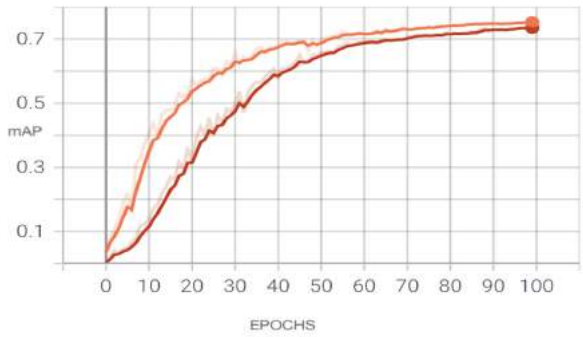

Fig. 3: Scaled YOLO v4 mean Average Precision for IoU 0.5 $mAP@0.5$ training over 100 epochs with pre-trained weights (orange graph), and random weights (red graph).

Performance of Scaled YOLO v4 Next, the performance of Scaled YOLO v4 is shown alongside the baseline model. Despite the different training setups and the reduced computational resources available for our model, our approach was able to achieved roughly a 10% higher mAP score than the baseline model, at a much faster inference speed of 9.2 ms per 480×480 image at batch size of 16.

Fig. 3 summarizes the development of the $mAP@0.5$ over 100 epochs for both a pre-trained and not pre-trained case. Since the use of pre-trained weights, represented by the orange line, achieves a higher accuracy, only this case is considered. One can observe a strong increase in accuracy in the first 25 epochs, then it improves at a slower rate. The best score achieved is a $mAP@0.5 = 75.5\%$, but could be even higher, as the model did not reach convergence. Further, it has to be noted that the baseline model addresses a simpler task than ours, that of object localization, where class labels were renamed to the same label.

Lastly, in Table I the performance on the testing set for each plant is summarized. These values can be biased, since a random testing set was chosen, so small deviations in the performance can be expected. The total amount of targets, which are the plant objects, is 1,300. The model reached an overall precision of 0.662, a recall of 0.749 and a $mAP@0.5$ of 0.745. It can be seen that there is a strong variety in performance scores. For example, CONAR has a mAP score of 97.8% and in contrast to this PPPMM, PPPDD and FUMOF all have a mAP that is less than 50%, with PPPMM being the worst at just 25.5%. PPPMM and PPPDD are the most difficult to detect, as they consist of either weeds, or the ending

TABLE I: Results of applying model to testing set

Class	Targets	P	R	mAP@.5
all	1330	0.662	0.749	0.745
ALOMY	525	0.629	0.657	0.68
ANGAR	1510	0.794	0.87	0.856
APESV	382	0.482	0.51	0.519
ARTVU	573	0.734	0.726	0.783
AVEFA	546	0.641	0.672	0.654
BROST	288	0.818	0.677	0.688
BRSNN	495	0.828	0.808	0.82
CAPBP	2670	0.771	0.764	0.763
CENCY	806	0.903	0.965	0.964
CHEAL	883	0.706	0.667	0.681
CHYSE	228	0.662	0.713	0.716
CIRAR	191	0.697	0.911	0.915
CONAR	178	0.833	0.972	0.978
EPHHE	22	0.243	0.818	0.755
EPHPE	330	0.902	0.861	0.884
EROCI	275	0.848	0.96	0.958
FUMOF	17	0.234	0.353	0.395
GALAP	321	0.915	0.903	0.918
GERMO	354	0.687	0.975	0.976
LAPCO	160	0.763	0.894	0.896
LOLMU	404	0.599	0.666	0.657
LYCAR	70	0.423	0.843	0.796
PPPDD	1500	0.398	0.484	0.431
PPPMM	559	0.375	0.302	0.255

leaves of plants that grew into a neighboring box. Similarly, FUMOF is the plant with the worst germination success and correspondingly with the least amount of target objects, which causes a very low mAP.

Fig. 4: Successful detection of GERMO plant objects.

Overall, the model achieved a very high mAP of over 97% for plants that grew well and where a rich data set existed, as Fig. 4 shows for the plant GERMO. This shows how the varying growth stages of plants can make this task more challenging. Moreover, the model achieves an almost 9% higher recall than precision. This means that the model is good at correctly detecting all existing plants, however, it occasionally detects a plant that does not exist. Little stones or small piles of soil could falsely be detected as tiny plants, hence creating false positives.

To further analyze the effect of the imbalance in the data set, a correlation matrix between the accuracy reached and the number of plant objects of each plant is plotted in Fig 5. It can be observed that the plants CIRAR, CONAR and GALAP all have a rather small amount of plant objects in total, but the model reaches a mAP of over 90% for all of them. Similarly, the plant EPHHE, which has the second least amount of plant objects, is still detected with a mAP of 75.5%. Interestingly, for this plant the model achieved a very low precision, but a rather high recall. This indicates that it identified too many instances as this specific plant, however, if the plant is in a given image, it detected the plant correctly. Although, the model performed very well on some minority classes, there is a slight positive correlation visible.

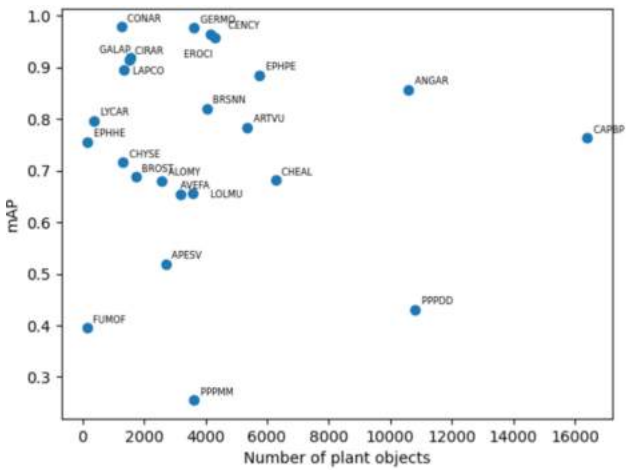

Fig. 5: This graph shows a scatter plot between the accuracy (mAP) reached and the number of plant objects of each plant

VI. Conclusion

In this research, a method based on SoA object detection is proposed for detecting plants in a novel data set, which stands out for its large size and high intra-class variability. In this dataset, diversity is provided by cultivating the plants under three different growth stages and by temporal tracking. This work improves upon the current baseline detection method for this data set, both in terms of accuracy and efficiency, by proposing appropriate data pre-processing steps and the implementation of a SoA real-time object detection model Scaled YOLO v4. The images were cropped to remove their black background, reducing the chance of false alarms, noise, while increasing resolution. This improved the models mAP by about 7%. Contrast was added to the images to enhance colors and edges, improving the model by 2%. Using pre-trained weights from the MS COCO data set, the model yielded a much faster increase in mAP and improved the model overall by roughly 2%. The performance of Scaled YOLO v4 is assessed on a representative sample of about half of the entire data set due to computational limitations. Nevertheless, the implemented model achieved an improvement of approximately 10% over the baseline plant localization model in terms of the mAP, while being much more efficient. Moreover, the proposed

model does not overfit and the consequence of the imbalanced data set is just minor. Finally, it can be expected that the model will achieve an even higher improvement when using all the available data and data augmentation to overcome data set imbalances.

REFERENCES

[1] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511015302476

[2] J. Möller, "Computer vision – a versatile technology in automation of agriculture machinery," 2010.

[3] R. Zwiggelaar, "A review of spectral properties of plants and their potential use for crop/weed discrimination in row-crops," *Crop Protection*, vol. 17, no. 3, pp. 189–206, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026121949800009X

[4] S. Tsaftaris, M. Minervini, and H. Scharr, "Machine learning for plant phenotyping needs image processing," *Trends in Plant Science*, vol. 21, p. 989–991, 12 2016.

[5] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Computers and Electronics in Agriculture*, vol. 178, p. 105760, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920312709

[6] S. L. Madsen, S. K. Mathiassen, M. Dyrmann, M. S. Laursen, L.-C. Paz, and R. N. Jørgensen, "Open Plant Phenotype Database of Common Weeds in Denmark," *Remote Sensing*, vol. 12, no. 8, p. 1246, 04 2020. [Online]. Available: https://www.mdpi.com/691100

[7] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," 2021.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[9] J. McKinion and H. Lemmon, "Expert systems for agriculture," *Computers and Electronics in Agriculture*, vol. 1, no. 1, pp. 31–40, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0168169985900043

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017, cite arxiv:1703.06870Comment: open source; appendix on more results. [Online]. Available: http://arxiv.org/abs/1703.06870

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[12] S. Bargoti and J. P. Underwood, "Deep fruit detection in orchards," *CoRR*, vol. abs/1610.03677, 2016. [Online]. Available: http://arxiv.org/abs/1610.03677

[13] J. Gao, A. P. French, M. P. Pound, Y. He, T. P. Pridmore, and J. G. Pieters, "Deep convolutional neural networks for image-based convolvulus sepium detection in sugar beet fields," *Plant Methods*, vol. 16, no. 1, p. 29, Mar 2020. [Online]. Available: https://doi.org/10.1186/s13007-020-00570-z

[14] M. O. Lawal, "Tomato detection based on modified yolov3 framework," *Scientific Reports*, vol. 11, no. 1, p. 1447, Jan 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-81216-5

[15] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," *CoRR*, vol. abs/1911.09070, 2019. [Online]. Available: http://arxiv.org/abs/1911.09070

[16] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019.

[17] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *CoRR*, vol. abs/1908.08681, 2019. [Online]. Available: http://arxiv.org/abs/1908.08681

[18] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *CoRR*, vol. abs/1911.08287, 2019. [Online]. Available: http://arxiv.org/abs/1911.08287