Deep learning based object detection for agricultural vehicles

Dimitrios Arapis
s193436
MSc in Autonomous Systems
3rd Semester Internship Project

***Abstract*—Object detection is a critical component as we move towards autonomous navigation of agricultural vehicles. Although numerous deep-learning based methods have been applied in the automotive and robotics industry, the challenges that appear in an agricultural environment need to be addressed. This research applies a state-of-the-art object detector, TinyYolov3, and evaluates its performance on detecting four common objects found in an agricultural field. For this purpose, two datasets have been created, one generated from online data and one from recordings in an agricultural field. The aim of this research was to extend current knowledge on methods to improve detection and attempt to reduce the time and effort of manually annotating the objects of interest. The experiments indicate that removing small annotated objects from the training data can improve the performance of the detector. Additionally, data augmentation operations on the original images such as (a) rotation, (b) histogram equalization and (c) rescaling have improved significantly the performance of the object detector, while no additional effort and time is required for annotations. More information including the trained model, testing images/videos as well as the tools used for data augmentation and size filtering are available online[1].**

***Index Terms*—object detection, agriculture, yolov3**

I. Introduction

In the last decades, there has been a constant development of technological equipment for agricultural vehicles which enables the improvement of operational efficiency, better management of crops and coordination between multiple vehicles. Various agricultural tasks previously performed by the operator such as seeding, planting, weeding and harvesting have been automated. In combination with auto-steering, the aforementioned technologies are considered key elements towards full tractor autonomy. However, the current state of autonomy is "assisted" and, therefore, requires an operator to monitor the autonomous process and interfere when needed. In order to further explore the possibility of full autonomous tractors, object detection and subsequently obstacle avoidance need to be addressed.

Over the last years, important improvements on that direction have been achieved mostly lead by the automotive industry. Despite the fact that some tasks and principles are similar in agriculture (e.g. safety of humans and animals, the vehicle itself and other surroundings), there are differences that need to be taken into consideration. In one way, a typical agricultural field is not as complex as the center of a modern city, however, the field environment introduces a big problem for object detection, which is heavy occlusions. Additionaly, the objects found in an agricultural field are different from the ones in a urban environment (e.g. wild animals, agricultural machinery) With the recent improvements in deep learning and the practical use of those techniques in autonomous cars, similar techniques are explored in an agricultural environment.

Dataset In order to apply and evaluate an object detection algorithm in agriculture, a representative dataset must be acquired. In opposition to the automotive industry, where a vast amount of existing and labelled datasets is accessible , in agriculture, similar datasets are not available. The few existing datasets are limited in size and have been created for specific research purposes, which makes them useful only for the same application (e.g. specific weed detection, counting citrus trees). Besides the size and availability limitations, existing datasets are usually of low quality, resulting in inefficient performance of the detector. Therefore, two new datasets were created for the purpose of this research.

Detection In contrast to the automotive scenario, an agricultural vehicle moves 5-10 times slower providing smaller stopping distances or more time to avoid an obstacle. As a result, being able to get closer to the object, a detector mounted on an agricultural vehicle can have more information on the object as it will appear bigger. Moreover, the problem of often occluded objects can be partially aided by having more instances of the object to be detected. Therefore, based on the aforementioned characteristics, TinyYolov3 was used as the object detection method.

Contribution The aim of this study was to create a dataset with some typical objects found in an agricultural environment and validate the feasibility of a state-of-the-art object detector in this scenario. In this context, an analysis of the factors related to the dataset that affect the performance of the detector was conducted. This work has highlighted the importance of filtering and augmenting the data used for training the object detector. Additionally, the tools created for this study are made available online for future use. The results of this study has so far been very encouraging and could be useful to improve knowledge about object detection for agricultural vehicles.

II. Related Work

Following the rising trend of Deep Learning (DL) based applications in all sectors, and mostly after 2014, different

[1] github.com/Dimitrios-Ar/agroObjectDetection

papers have been published utilizing its tools in agriculture [16]. The most researched DL agricultural applications are phenology recognition, disease detection, crop classification, weed detection, fruit counting and yield prediction.

Yalcin proposed Deep-Pheno for the discrimination of the different phenological stages of the plants [43]. Bauer et al. used a method for large-scale phenotyping using NDVI aerial images [3]. Taghavi et al. [33] and Pound et al. also proved that DL features perform better than hand-crafted features for plant phenotyping [24].

Sladojevic et al. introduced a model able to recognize different types of plant diseases on leaves using the Caffe deep learning framework achieving precision between 91-98% [31]. Amongst others, Barbedo also showed the power of Convolutional Neural Networks (CNNs) for plant disease classification but addressed their practical limitation which is lack of data [2]. Too et al. conducted a comparative study of deep learning architectures for disease detection scoring 99.75% on DenseNet [36]. Thenmozhi et al. used CNNs for crop pest classification with a 95.97% accuracy in a task that has been proven challenging for farmers [34].

Yu et al. tested four different CNN algorithms for single and multiclass weed detection [44]. Farooq et al. compare CNN and Histogram of Oriented Gradients (HOG) methods for weed classification and prove that CNNs provide more accurate classification [11]. Kounalakis et al. also proposed a deep learning based weed recognition system demonstrating its effectiveness againts handcrafted feature based methods [17].

Grinblat et al. use leaf vein patterns as input to a CNN to classify three different species and prove that the accuracy increases as the model deepens [12]. Zhu et al. used a pre-trained AlexNet network to classify normal and abnormal carrots aiming in quality control achieving 98,7% accuracy [46].

Chen et al. applied R-CNN to count strawberries from RGB images captured with the use of a UAV flying 2 to 3 meters above the plants [6]. Csilik et al. also used UAV images and CNNs for identification of citrus trees [9]. Häni et al. compare their DL approach to counting apples with Gaussian Mixture Models (GMM) and prove that DL outperforms the GMM in three out of four datasets, even though GMM parameters are tuned for each dataset [14]. Rahnemoonfar et al. used a variation of Inception-Resnet to count fruits and estimate the yield by using synthetic image resulting in an algorithm that can handle occlusions and is robust on scale and illumination variations with an accuracy of 91% [25]. Silver et al. investigated different methods for DL based estimation of the weight of grapes from a smartphone image [30]. Tian et al. used Yolov3 amongst others for a comparison of DL based and traditional methods for detection of apple lesions [35].

Steen et al. explored the use of CNN to detect a specific barrel-shaped object in different scenarios including occlusions and documented high performance [32]. Christiansen et al. proposed a combination of background subtraction and deep learning techniques which deliver high accuracy in detecting obstacles especially in far distances when compared with other detectors including Yolo [7].

In the general framework of object detection, Wu et al. addressed the issue that appears due to the downsizing of input images resulting in losing too much information and therefore being unable to recognize the object [40]. In the task of object detection, different solutions targeted on the CNNs have been applied to improve the performance of specific detectors for small scale objects ([5],[4],[13],[27],[21],[26]).

The importance of data augmentation to improve the task of object recognition and also assist on small scale object detection has been broadly addressed ([28],[47],[40]). Zoph et al. found that data augmentation improves the performance of the detector especially on small datasets and on detecting small objects [47]. Some of the most common data augmentation techniques include operations (a) on the whole image, (b) on part of the image and (c) synthetic data. Also, techniques applied on the whole image or part of the image can be categorized in (a) color operations, (b) geometric operations and (c) bounding box operations [47]. Zagoruyko et al. applied horizontal flips and cropped randomly the images using reflections of the original image to fill the missing pixels [45]. In [19], each training image is randomly flipped, cropped or distorted. Ciresan et al. used translation operation on images to improved the error from 28% to 20% on MNIST dataset [8]. Paulin et al. proposed a method for greedily selecting the best augmentation operation that shows the highest accuracy gain at each iteration [23]. Lamley et al. proposed a network that generates augmented data to reduce network loss while training by combining samples from the same class [18]. Nilsson et al. used generated images combining rendered virtual pedestrians on real backgrounds leading in better performance compared to using only real training data [20]. Vazquez et al. also generated virtual data and combined them with real world data to delivered slightly worse performance than the one achieved with the double manually labeled dataset [38].

While recent work has shown that data augmentation improves the performance of the detector, there are some cases where it is stated otherwise. Especially in [10], Cubuk et al. showed that the performance improved while adding more data augmentation operators up to 20, but after did not show any significant improvement. In contrast, worse results appear when completely randomizing the probabilities and magnitudes of each operator, the detector under-performed.

III. Methods

For Yolov3 and similar CNN-based object detection, a vast amount of annotated images is required. Available public datasets contain annotations of some classes, however, for the majority of other tasks, the researcher need to provide annotated images for the detector. This task is an active research topic during the last decade, with different provided solutions improving the time and effort required ([22],[37]).

Four classes were selected as targets of detection. These classes are (a) trees, (b) utility vehicles, (c) tractors and (d) people and can be seen in the left column of Figure 2. The selection was based on the idea of having variations such as

(a) sizes, (b) static and dynamic objects and (c) shape but also analyze the performance for more resembling objects (tractor and utility vehicle).

The targeted scenario was to explore the performance of the detector in a real scenario, and therefore, one dataset was created by using field recordings captured on September 2019 in the area of Randers, Denmark. From the recordings, 5 frames per second at a resolution of 2048x1536 pixels were extracted. From a total of more than 20000 images, most not containing any object, 2525 were selected. However, a detector trained on a dataset consisting of only one instance of a specific class (e.g. one tractor model) would lead to overfitting within the specific dataset and underperform on most others. Therefore, a second dataset using online data was generated. The online dataset was created using public data by searching for each of the aforementioned classes. A total of 3882 images established the online dataset. For each image on both datasets, the researcher manually annotated the objects. Throughout this paper, the terms 'field' and 'online' will be used to distinguish the two datasets.

Hardware The field dataset was created using the Blackfly S GigE 3.2MP sensor mounted on a combine harvester. The system used during all steps including manipulating data, creating additional tools based on the object detector, obtaining online dataset, training and testing was an Ubuntu 18.04, Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz with 16GB of RAM equipped with a AORUS GeForce GTX 1080 8GB GPU.

Annotation Process The Computer Vision Annotation Tool (CVAT) developed by Intel was used for annotating both datasets ([15],[29]).

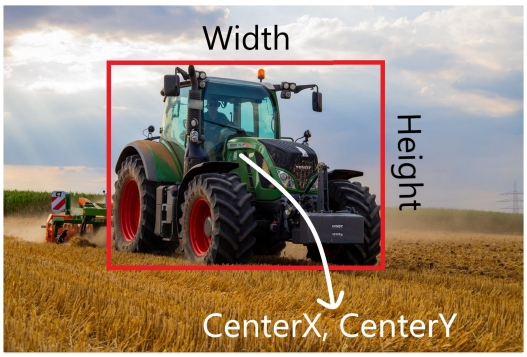
Detector The main purpose of this project was the creation of the dataset and different strategies to accelerate and improve this procedure. Therefore, the detector and parameters were decided in the beginning of the project and did not change when applying other strategies. For the initial selection of the detector, a performance test of three considered variations of the Yolo framework was conducted. More specifically, the results of the performance of (a) TinyYolov3 416x416, (b) TinYolov3 608x608 and (c) Yolov3 416x416 will be presented in Section III. The difference of (a) and (b) is the input size of the image, being 416x416 and 608x608 respectively. The dataset is the same, as the rescaling of each image happens by the detector before it is used as the input. In Figure I, the architecture of the chosen algorithm is presented.

Following the relevant documentation [1], the detector needs at least 2000 iterations per class (8000 for 4 classes), which lead to a selection of 10000 iterations per training. For all trainings the batch size was 64 and the subdivisions 8. The learning rate was fixed at 0.001 from iteration 0 to 8000 and then reduced twice, once at 8000 iterations and one at 9000 iterations, by a factor of 10.

Size filtering For each annotated object, a bounding box exists with stored information of the Class_id, bounding box center x, bounding box center y, width and height. The dimensional data is normalized depending on the image width and height, resulting in a number in the range [0,1]. The exact format is visualized in Figure 1. The bounding box width and height are used to calculate the coverage of each object in the image. During size filtering, any object with a bounding box smaller than a set threshold of total pixels is removed.

Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	3x3/1	608x608x3	608x608x16
1	Maxpool		2x2/2	608x608x16	304x304x16
2	Convolutional	32	3x3/1	304x304x16	304x304x32
3	Maxpool		2x2/2	304x304x32	152x152x32
4	Convolutional	64	3x3/1	152x152x32	152x152x64
5	Maxpool		2x2/2	152x152x64	76x76x64
6	Convolutional	128	3x3/1	76x76x64	76x76x128
7	Maxpool		2x2/2	76x76x128	38x38x128
8	Convolutional	256	3x3/1	38x38x128	38x38x256
9	Maxpool		2x2/2	38x38x256	19x19x256
10	Convolutional	512	3x3/1	19x19x256	19x19x512
11	Maxpool		2x2/2	19x19x512	19x19x512
12	Convolutional	1024	3x3/1	19x19x512	19x19x1024
13	Convolutional	256	1x1/1	19x19x1024	19x19x256
14	Convolutional	512	3x3/1	19x19x256	19x19x512
15	Convolutional	27	1x1/1	19x19x512	19x19x27
16	Yolo				
17	Route 13				
18	Convolutional	128	1x1/1	19x19x256	19x19x128
19	Upsampling		2x2/1	19x19x128	38x38x128
20	Route 19 8				
21	Convolutional	256	3x3/1	38x38x128	38x38x256
22	Convolutional	27	1x1/1	38x38x256	38x38x27
23	Yolo				

TABLE I: TinyYolov3 608x608 Architecture.



Class_id	CenterX	CenterY	Width	Height
Tractor	0,468122	0,462354	0,533496	0,590901

Fig. 1: Yolov3 bounding box components.

Data augmentation The data augmentation strategy followed was to use three operations on the data. The selected (a) rotation and (b) equalize were found the two most beneficial operations in [47] and (c) rescale is also commonly utilized in [41] and [39] and was used in an attempt to fill the observed gap of small objects in the online dataset. This observed gap can be explained as a consequence of the object-centric search of images.

For the rotation operation, a random rotation in the range of (-30,30) degrees was applied. The equalize operation was handled using contrast limited adaptive histogram equalization (CLAHE) - (contrast limit=2.0, tile size=(8,8)). Finally, during the rescale operation the original image was downscaled to

between 70 to 90 % of its original size, and randomly positioned within a black background of its original size.

During these operations, the necessary modifications of the annotation files were made to ensure the correct displacement of the bounding box in order to maintain consistency in the data. As also mentioned in [47], during the rotation operation area covered by the bounding box is increased (see Figure2, column: Center right).

Fig. 2: Data augmentation strategy. Left: Original image, Center left: Equalized image, Center right: Rotated image, Right: Rescaled image

IV. Results

Detector

	TinyYolov3 416x416	TinyYolov3 608x608	Yolov3 416x416
mAP@0.5	68.12%	**75.24%**	**76.15%**
Speed	**110 fps**	**100 fps**	55 fps
Training time	**2.5 hours**	**4 hours**	>20 hours

TABLE II: Characteristics of Yolov3 variations

Size filtering It was decided that the best procedure for this investigation was to limit factors that could not be monitored. For that reason, the field dataset was used for this experiment as it only contains multiple instances of the same object. Therefore we hypothesized that filtering out data can mainly be reflected on the object size and not on the fact that the detector loses crucial information of the object (e.g. specific angle). During the different thresholds, the amount of data filtered out was in the range of 0 to 5.3% of the total dataset, and, thus, combined with findings [42] - "dropping 30% correct annotations result in a drop of only 5% mAP" - also leads to the hypothesis that a substancial change in performance is mostly affected by the size of the filtered annotation and not by losing information on the specific object. However, due to the changes on the training data, these findings need to be interpreted with caution.

In an attempt to identify how small sized object affect TinyYolov3, the following experiment was designed. Our approach was to train the algorithm using 70% of the corresponding size filtered data and test on two testing sets, one consisting of the remaining 30% of the filtered data and the other of the same 30% unfiltered.

The first group was automatically formed with unfiltered training and testing data. Then, five more groups were formed, one for each threshold:

- Filtered training data for threshold x
- Filtered testing data for threshold x

The thresholds used were 200,400,600,800 and 1000 pixels.

As expected, by filtering both training and testing data, the mAP@0.5 constantly increases (see Figure 3). Especially by just removing the objects smaller than 200 pixels, which translates to a removal of <2% of the data, a rise of 6.3% on mAP@0.5 was observed. However, the most remarkable result to emerge from the data is the performance on detecting objects on unfiltered data. Again, the biggest rise is visible when filtering out objects smaller than 200 pixels. In this case though, results show that the algorithm performs better on detecting small objects when it is not trained on small objects. The maximum performance of unfiltered testing was achieved for a threshold of 600 pixels. When applying bigger thresholds, the mAP@0.5 starts to drop again which would indicate that at this point the algorithm starts losing information crucial to detect the same objects in bigger distances.

For the remaining experiments and results all data were filtered and did not contain any object with a bounding box smaller than 600 pixels, unless stated otherwise.

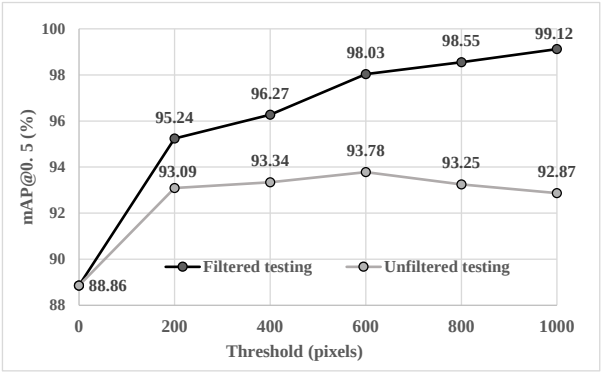


Fig. 3: Mean Average Precision for different size filtering thresholds.

Data augmentation From the total of 2525 images of the online dataset containing in total 3229 annotated objects (868 trees, 663 utility vehicles, 936 tractors, 762 people), a random selection was made to obtain 200 annotations per class. Using the selected 453 images, one original and five augmented training sets were created:

- ORI: Original images. Total images: 453
- ROT: All images have been rotated. Total images: 906

- EQU: All images have been equalized. Total images: 906
- RES: All images have been rescaled. Total images: 896
- RAN: A random operation is applied into each image (roughly 1/3 images are rotated, 1/3 equalized and 1/3 resized) and all images are added to the original images. Total images: 906
- ALL: A combination of the original images and all operations. Total images: 1802

In the RES set, due to the results of size filtering section, if the rescaled object was smaller than 600 pixels, the annotation would not be used.

Using the aforementioned six training sets, the performance of the detector was tested on two testing datasets, one consisting of the remaining images from the online dataset and the other one being the complete field dataset. As seen in Table III, the performance of the detector increased by adding rotated, equalized and rescaled images by any amount and combination tested on both training sets.

Training	Testing Online	Testing Field
ORI	71.42%	28.36%
ROT	71.82%	30.63%
EQU	72.31%	32.22%
RES	73.01%	**35.26%**
RAN	72.26%	31.15%
ALL	**73.27%**	34.64%

TABLE III: Mean average precision (mAP@0.5) comparison for each set of training and testing set.

Interestingly, most operations appear to have similar affect on the performance of the detector on both datasets . Compared to the ORI set, the smallest increase was found when using ROT set, followed by bigger increase when using EQU set and finally showing the best results when using RES set. Randomly applying one of the operations to each original image (RAN) was more effective than ORI and ROT, but not as effective as EQU or RES. However, different behavior can be observed on the best strategy to detect the objects of the online dataset and those of the field dataset. For the first, combining all operations (ALL) leads to the highest increase whereas for the second, adding only the rescaled images proves to work better. These results are in line with previous results [10] which also show that there are cases when adding augmentation operations that worsen the results. Interestingly, the magnitude of the performance increase was bigger on the field data which was generally worse detected. Compared to the original data, by adding randomly rescaled images, the performance on the online dataset jumped from 28.36% to 35.26%.

V. Conclusion

This work has proved that a state-of-the-art object detector, TinyYolov3, can be used to detect objects in an agricultural environment. Despite the fact that there are limitations due to the usage of a specific object detection and input size of the images, this study has highlighted the importance of considering filtering the annotations based on the size of the annotated object. By removing small objects from the training data, an increase in the range of 4-5% appeared. Additionally, the evidence from this study suggests that using data augmentation, without the need of any additional time and effort to annotate new images, can increase the performance of the detector up to 24%. The results of Table III give some insights on how each augmentation strategy affects its performance, however some differences on how it affects each dataset are visible. Therefore, a general rule on which augmentation to use for object detection cannot be established, as it remains strongly connected to the dataset provided and the gap between training and testing data.

Further work needs to be carried out to establish a generic framework to help future object detection applications with detailed augmentation strategies. On a wider level, research is also needed to address the problem of limited data on some agricultural objects either by introducing a synthetic data generator tool or by introducing new methods of detecting obstacles with less need of training data.

References

[1] AlexeyAB. darknet. https://github.com/AlexeyAB/darknet, 2019.

[2] J. G. A. Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46 – 53, 2018.

[3] A. Bauer, A. Bostrom, J. Ball, C. Applegate, T. Cheng, S. Laycock, S. Rojas, J. Kirwan, and J. Zhou. Combining computer vision and deep learning to enable ultra-scale aerial phenotyping and precision agriculture: A case study of lettuce production. *HORTICULTURE RESEARCH*, 6, 06 2019.

[4] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu. An improved faster r-cnn for small object detection. *IEEE Access*, 7:106838–106846, 2019.

[5] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu. Feature-fused SSD: fast detection for small objects. *CoRR*, abs/1709.05054, 2017.

[6] Y. Chen, W. S. Lee, H. Gan, N. Peres, C. Fraisse, Y. Zhang, and Y. He. Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages. *Remote Sensing*, 11(13), 2019.

[7] P. Christiansen, L. Nielsen, K. Steen, R. Jørgensen, and H. Karstoft. Deepanomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, 16(11):1904, Nov 2016.

[8] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[9] O. Csillik, J. Cherbini, R. Johnson, A. Lyons, and M. Kelly. Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones*, 2(4), 2018.

[10] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] A. Farooq, J. Hu, and X. Jia. Analysis of spectral bands and spatial resolutions for weed classification via deep convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 16(2):183–187, Feb 2019.

[12] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto. Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture*, 127:418 – 424, 2016.

[13] G. Y. Huang Jipeng, Shi Yinghuan. Multi-scale faster-rcnn algorithm for small object detection. *Journal of Computer Research and Development*, 56(2):319, 2019.

[14] N. Häni, P. Roy, and V. Isler. Apple counting using convolutional neural networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2559–2565, Oct 2018.

[15] Intel. cvat. https://github.com/opencv/cvat, 2019.

[16] A. Kamilaris and F. Prenafeta Boldú. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, pages 1–11, 06 2018.

[17] T. Kounalakis, G. A. Triantafyllidis, and L. Nalpantidis. Deep learning-based visual recognition of rumex for robotic precision farming. *Computers and Electronics in Agriculture*, 165:104973, 2019.

[18] J. Lemley, S. Bazrafkan, and P. Corcoran. Smart augmentation - learning an optimal data augmentation strategy. *CoRR*, abs/1703.08383, 2017.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[20] J. Nilsson, P. Andersson, I. Y. H. Gu, and J. Fredriksson. Pedestrian detection using augmented training data. In *2014 22nd International Conference on Pattern Recognition*, pages 4548–4553, Aug 2014.

[21] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng. R^2-cnn: Fast tiny object detection in large-scale remote sensing images. 02 2019.

[22] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari. Extreme clicking for efficient object annotation. *CoRR*, abs/1708.02750, 2017.

[23] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. Transformation pursuit for image classification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3646–3653, June 2014.

[24] M. Pound, A. Burgess, M. Wilson, J. Atkinson, M. Griffiths, A. Jackson, A. Bulat, Y. Tzimiropoulos, D. Wells, E. Murchie, T. Pridmore, and A. French. Deep machine learning provides state- of-the-art performance in image-based plant phenotyping. *GigaScience*, 6, 05 2016.

[25] M. Rahnemoonfar and C. Sheppard. Deep count: Fruit counting based on deep simulated learning. *Sensors (Basel, Switzerland)*, 17, 04 2017.

[26] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[27] Y. Ren, C. Zhu, and S. Xiao. Small object detection in optical remote sensing images via modified faster r-cnn. *Applied Sciences*, 8:813, 05 2018.

[28] I. Sato, H. Nishimura, and K. Yokoi. Apac: Augmented pattern classification with neural networks. 05 2015.

[29] N. M. A. Z. Sekachev, Boris. Computer vision annotation tool: A universal approach to data annotation. https://software.intel.com/en-us/articles/computer-vision-annotation-tool-a-universal-approach-to-data-annotation, 2019.

[30] D. Silver and T. Monga. In vino veritas: Estimating vineyard grape yield from images using deep learning. 03 2019.

[31] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. In *Comp. Int. and Neurosc.*, 2016.

[32] K. Steen, P. Christiansen, H. Karstoft, and R. Jørgensen. Using deep learning to challenge safety standard for highly autonomous machines in agriculture. *Journal of Imaging*, 2:6, 02 2016.

[33] S. Taghavi, M. Esmaeilzadeh, M. Najafi, T. Brown, and J. Borevitz. Deep phenotyping: Deep learning for temporal phenotype/genotype classification. *Plant Methods*, 14, 12 2018.

[34] K. Thenmozhi and U. S. Reddy. Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*, 164:104906, 2019.

[35] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang. Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense. *J. Sensors*, 2019:7630926:1–7630926:13, 2019.

[36] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161:272 – 279, 2019. BigData and DSS in Agriculture.

[37] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, and V. Chari. Learning to generate synthetic data via compositing. *CoRR*, abs/1904.05475, 2019.

[38] D. Vazquez, A. M. Lopez, and D. Ponsa. Unsupervised domain adaptation of virtual and real worlds for pedestrian detection. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3492–3495, Nov 2012.

[39] A. Visser and J. van Gemert. Content-aware image resizing for faster object detection on aerial imagery. *Paper presented at Small UAS for Environmental Research Conference*, 2016.

[40] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *ArXiv*, abs/1501.02876, 2015.

[41] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[42] Z. Wu, N. Bodla, B. Singh, M. Najibi, R. Chellappa, and L. S. Davis. Soft sampling for robust object detection. *CoRR*, abs/1806.06986, 2018.

[43] H. Yalcin. Plant phenology recognition using deep learning: Deep-pheno. In *2017 6th International Conference on Agro-Geoinformatics*, pages 1–5, Aug 2017.

[44] J. Yu, A. W. Schumann, Z. Cao, S. M. Sharpe, and N. S. Boyd. Weed detection in perennial ryegrass with deep learning convolutional neural network. *Frontiers in Plant Science*, 10:1422, 2019.

[45] S. Zagoruyko and N. Komodakis. Wide residual networks. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016.

[46] H. Zhu, L. Deng, D. Wang, J. Gao, J. Ni, and Z. Han. Identifying carrot appearance quality by transfer learning. *Journal of Food Process Engineering*, 42(6):e13187, 2019.

[47] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens, and Q. V. Le. Learning data augmentation strategies for object detection. *CoRR*, abs/1906.11172, 2019.

Appendix

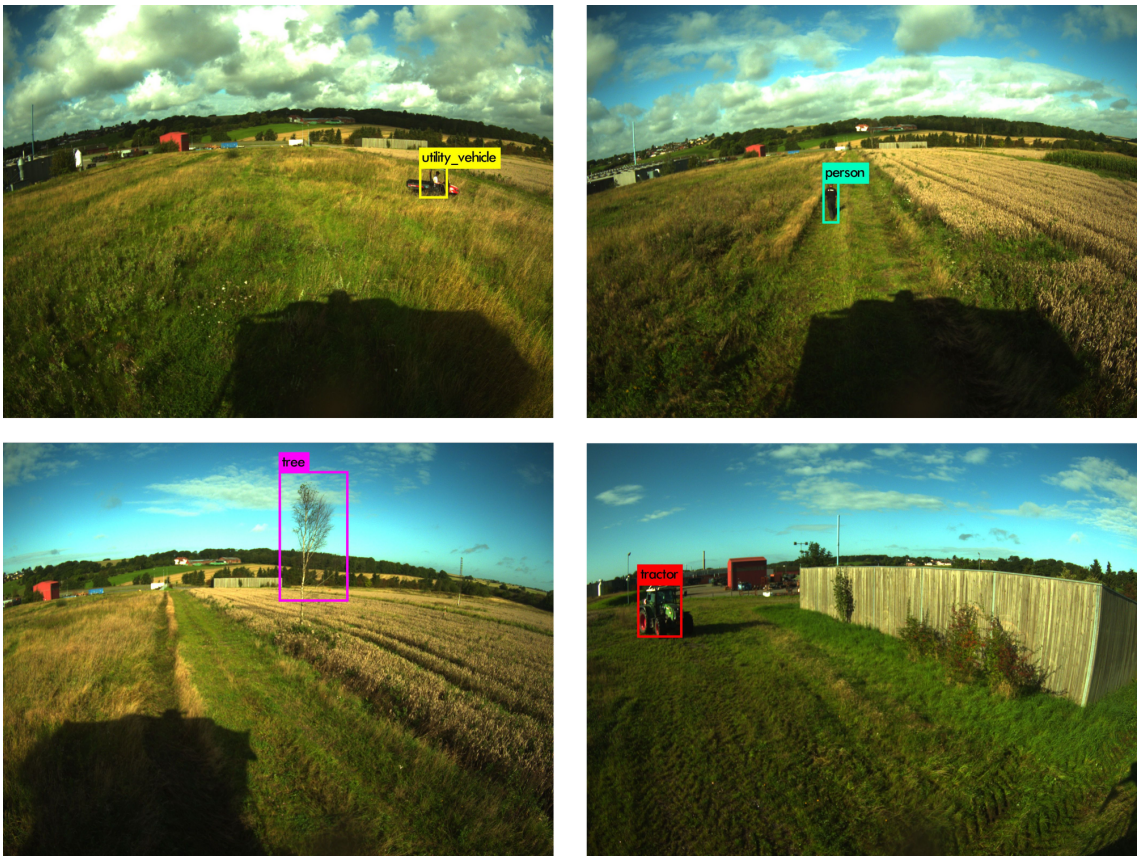


Fig. 4: Example of field detections. Detector is trained using the online dataset.