The pitfalls of transfer learning in computer vision for agriculture

Julius Autz[1], Saurabh Kumar Mishra[1], Lena Herrmann[1] and Joachim Hertzberg[1,2]

Abstract: Computer vision applications based on modern AI methods are becoming increasingly important in agriculture, supporting and automating common processes. These applications are usually based on well-established architectures and pre-trained models. However, our prior experience has shown that applying the concept of transfer learning to AI tasks in agriculture repeatedly resulted in systematic issues. The structure of agricultural images, containing objects similar in shape, color and texture, makes the reuse of well-established applications more challenging. To give a more detailed insight into the expected challenges, we trained two different networks, which are well-established in the literature: Mask R-CNN and YOLOv5 [He18; Jo21] and investigated them in two different learning setups. First, we applied the concept of transfer learning to these models by pre-training each on the COCO dataset and subsequently continued expanding the available target set with classes of the sugar beets dataset [Ch17]. In the second setup, we skipped pre-training and only trained the models on the given agriculture dataset. Furthermore, we describe the reasons for the results in more detail and highlight possible causes for the identified differences. Finally, the different performances of the networks allowed us to improve on best practices for the agricultural domain and give some advice for future computer vision tasks in this area.

Keywords: Machine Learning, Precision Farming, Transfer Learning Computer Vision

1 Introduction

Machine learning, and by extension ML-based computer vision techniques have been successfully applied in numerous areas [Ch17], recently expanded by applications in the area of agriculture [Li18]. There, they enable the automation, and therefore simplification, of existing processes like tracking crop growth, monitoring livestock or detecting damaged fruits during harvest. Aside from saving manpower and costs, obvious interests of the industry, they are also capable of increasing efficiency and sustainability, e.g. by reducing waste [Ba18].

The large amount of data required to develop reliable automated processes can be overcome with the application of transfer learning, where a network previously trained on a different dataset is used to learn new classes of interest [We16]. For example, within object detection, transferring global features such as edges can help reduce training time [Yo14] and improve results when data is limited or difficult to acquire. Having the

[1] DFKI Labor Niedersachsen, Plan Based Robot Control, Osnabrück, vorname.nachname@dfki.de
[2] Osnabrück University, KBS Group, Osnabrück, joachim.hertzberg@uni-osnabrueck.de

challenges of computer vision in agriculture in mind, these arguments for transfer learning seem to apply directly for this area of work. At a glance, reusing a pre-trained neural network should lead to qualitative reliable results. For transfer learning to work, however, the source and target domain need to be sufficiently related to successfully transfer the already learned structures to new images.

Previous experience shows that these techniques repeatedly failed when transferred into the domain of agriculture. Therefore, we decided to directly compare two networks in the manner of transfer learning and end-to-end training, to show that transfer learning is not always a valid option to achieve representative results. We based our experiments on Mask-RCNN and YOLOv5 [He18] [Jo21], using the sugar beets dataset as a representative of the agricultural domain [Ch17]. We analyzed the networks pre-trained on the COCO-dataset [Li14] and without any previous knowledge. A comparison of the results will show that the application of transfer learning is not as beneficial as would be assumed in the first place. To the best of our knowledge, there has not been a comparison between transfer learning and training from scratch using established neural network architectures and publicly available datasets for the agricultural domain. Our research seeks to address this gap of experiments and provide a better understanding of the limitation and requirements of transfer learning in this specific context, across multiple networks.

2 Related Work

Other works have used neuronal networks in various agricultural contexts, such as plant phenotyping [Si16], fruit detection and yield estimation [BU17], plant health monitoring [Mo14] and more [Li18]. In these cases, enough data has been (manually) labeled that the networks can be trained from scratch and provide good performance. However, in many agricultural use cases it is challenging to collect sufficient training data. Here, transfer learning can ease the burden by reducing the amount of training data needed for new use-cases [We16]. Transfer learning is an active research topic, with applications and research ranging from medical imaging/disease recognition to traffic scenes/driving assistance [Zh20] and beyond. In the field of agriculture, several recent experiments have used transfer learning with success [Bo20]. At the same time, as transfer learning is applied more often, its limitations and prerequisites are revealed, leading to analyses of so called ”negative transfer” [Wa19; We16], supplementing and expanding upon prior research on the limitations of transfer learning [Yo14].

3 Models overview

The Mask R-CNN [He18] belongs to the family of R-CNN architectures, extending from Faster R-CNN. It is a two-stage process; the first stage is the Region Proposal Network (RPN), where the category-independent region proposals are generated from an image.

The second stage consists of object classification, bounding box regression and mask-prediction. The paper's authors used variants of ResNets as a backbone and performed an ablation study with and without Feature Pyramid Network (FPN). The current state of the art in instance segmentation architecture – Swim Transformer [Li21] in terms of mean average precision (mAP) metric on COCO dataset [Li14], also used Mask R-CNN with Swim-T as a backbone. Further, we use YOLOv5, the latest version of the YOLO object detector and closely based on YOLOv4 [Jo21; Bo20]. With YOLOv4, new features were introduced to further improve the accuracy and performance of the network. Among those features are the Mosaic and Self-Adversarial Training for data augmentation, which provide a mechanism to let the model learn to detect an object independent of its context. Through the development of YOLOv5, the model was transferred from the Darknet framework to pytorch [So21]. Furthermore, according to the authors, the architecture is now more suitable for smaller edge devices with still good performance. This makes the YOLOv5 a proper choice for an application within the agriculture area, as farming robots or even larger machines only provide little space for an operating unit.

4 Experiments

We use two different network architectures: Mask R-CNN and YOLOv5. Both have proven useful within the areas of object detection and segmentation. We trained each network in two different settings. In the first setup, we used each network pre-trained on the COCO dataset [Li14] and transferred it to the new domain. In the second setup, we trained each network end-to-end from scratch. As our target we used the sugar beets dataset [Ch17], containing plants and weeds, as an object detection task related to precision farming. The images are available in RGB and near infrared with corresponding segmentation masks, but we only focused on the RGB images because they are commonly used and were also the base of the COCO dataset. The dataset contains about 11.000 images, split into four smaller parts containing 1.000, 2.000, 3.000 and 5.000 images for training. In addition to that, we selected fixed parts for testing with 1.000 and validation with 2.000 images, keeping those parts entirely distinct.

The reduced datasets resemble situations where data is limited and transfer learning is a valid option. During a pre-processing step we first reduced the size of the images to 512 x 512 and further transformed the segmentation masks into the COCO and the YOLOv1.1 format, respectively. Each network was trained with optimized hyperparameters, for different amounts of epochs, depending on the network and the training behavior. We trained the YOLOv5 for 300 epochs and the Mask R-CNN for 8.000+ epochs.

5 Results

Comparing the networks with each other is difficult, given that Mask RCNN unlike YOLOv5 uses segmentation masks in addition to bounding boxes. We compare the performance within a network across the different sized datasets, as well as pre-trained and training from scratch. For Mask R-CNN, we use Average precision (AP) as metric to evaluate the performance of object detection and segmentation. The calculation is based on the intersection over union (IoU) with different levels of thresholds. AP^{50} and AP^{75} calculating the precision over all classes with a threshold of 50 % and 75 % respectively. The metric $AP^{50:05:95}$ applies 10 thresholds ranging from 50 % to 95 % with a step size of 5 % between each value. The mAP used for YOLOv5 is calculated after the same principle, but expressed as percentage values.

Mask R-CNN	**1k Datapoints**		**2k Datapoints**		**3k Datapoints**		**5k Datapoints**	
Scratch	bbox	segm	bbox	segm	bbox	segm	bbox	segm
$AP^{.50:.05:.95}$	51.40	49.11	54.03	51.74	54.29	52.26	54.81	52.20
$AP^{.50}$	71.57	71.95	73.90	74.39	74.65	74.91	74.91	75.33
$AP^{.75}$	56.19	55.15	58.57	57.85	59.28	58.66	60.13	58.43
Pre-trained								
$AP^{.50:.05:.95}$	52.24	53.59	53.59	50.84	53.59	51.24	53.83	51.47
$AP^{.50}$	73.17	73.71	73.53	73.87	73.74	74.12	73.95	74.55
$AP^{.75}$	59.24	58.72	60.93	59.77	60.74	60.04	61.09	60.08

Tab 1: Performance overview of Mask R-CNN

In case of the Mask R-CNN, we can observe in Table 1 that more data leads to better performance for both the pre-trained and the end-to-end training. However, the pre-trained network solely achieves higher metrices when applying 1k datapoints. This holds true for bounding boxes and segmentation alike. If the amount of data increases, the training from scratch constantly outperforms the pre-trained variation. As such, one can consider there to be a negative transfer, where the prior knowledge of the COCO dataset hinders the learning of the new sugar beets dataset. This becomes apparent as soon as the amount of sugar beet training data is sufficient. Yet if we apply a higher threshold, like for $AP^{.75}$, the pre-trained network shows slightly better results. In summary, no training significantly outperforms the other, suggesting there is no substantial benefit from transfer learning in this application.

YOLOv5				
Scratch	**1k Datapoints**	**2k Datapoints**	**3k Datapoints**	**5k Datapoints**
$mAP^{.50}$	0.76	0.77	0.81	0.75
$mAP^{.50:.05:.95}$	0.55	0.58	0.62	0.55
Pre-trained				
$mAP^{.50}$	0.73	0.75	0.77	0.79
$mAP^{.50:.05:.95}$	0.48	0.50	0.55	0.57

Tab 2: Performance overview of the YOLOv5

The results of the YOLOv5, while not as straightforward to interpret in regard to the amount of data, still showed better performance for the scratch trained network, except in the case of 5k datapoints (Table 2). In this case, the performance of the network abruptly drops. The pre-trained YOLOv5 used a special variation of transfer learning wherein the weights of the backbone are frozen for 250 epochs and reintegrated into the training for the last 50 epochs. Although we apply the pre-trained YOLOv5, which should lead to better results, it still underperformed compared to the end-to-end training, which indicates negative transfer.

6 Conclusion

While transfer learning remains a useful tool in many cases, this paper shows that blind application of pretrained networks not only does not improve performance, but may instead hinder it. Especially in situations where there are not immense, but still substantial amounts of data, a more efficient network trained from scratch seems to be a better solution than an extensively pretrained large one. While the performance often can be equalized by extensive re-training, the ”unlearning” of the inapplicable information, combined with the pre-training and the related computational requirements cause additional costs. Until data becomes as broadly available for agriculture as it has become for other fields, we recommend using transfer learning sparsely, between related fields, and avoiding large networks which require equally large datasets to train.

The DFKI Niedersachsen Lab (DFKI NI) is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung.

References

[Bo20] Bosilj,P.; Aptoula, E.; Duckett, T; Cielniak, G.: Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics*, 37(1):7-19, 2020.

[Ba18] Balducci F.; Impedovo, D.; Pirlo, G.: Machine learning applications on agricultural datasets for smart farm enhancement. *Machines*, 6(3):38, 2018.

[BU17] Bargoti, S.; Underwood J.: Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039-1060, 2017.

[Bo20] Bochkovskiy, A.; Wang, C. Y.; Liao, H.: Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[Ch17] Chebrolu, N.; Lottes, P.; Schaefer, A.; Winterhalter, W.; Burgard, W.; Stachniss, C: Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 2017.

[Da20] Dargan, S.; Kumar, M.; Ayyagari, M.; Kumar, G.: A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071-1092, 2020.

[He18] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.: Mask r-cnn, 2018.

[Jo21] Jocher, G.; et. al: ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, April 2021.

[Li18] Liakos, K.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D.: Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018.

[Li21] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[Li14] Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick C.: Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740-755. Springer, 2014.

[Mo14] Moshou, D.; Pantazi, XE.; Kateris, D.; Gravalos, I.: Water stress detection based on optical multisensor fusion with a least squares support vector machine classifier. *Biosystems Engineering*, 117:15-22, 2014.

[Si16] Singh, A.; Ganapathysubramanian, B.; Singh, A. K.; Sarkar, S.: Machine learning for high-throughput stress phenotyping in plants. *Trends in plant science*, 21(2):110-124, 2016.

[So21] Solawetz, J.: Yolov5 new version - improvements and evaluation. https://blog.roboflow.com/yolov5-improvements-and-evaluation/, accessed: 21.10.2021.

[Wa19] Wang, Z.; Dai, Z.; Póczos, B.; Carbonell, J.: Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11293-11302, 2019.

[We16] Weiss, K.; Khoshgoftaar, T.; Wang, D.: A survey of transfer learning. *Journal of Big data*, 3(1):1-40, 2016.

[Yo14] Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H.: How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[Zh20] Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q.: A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43-76, 2020.