# Development of an intelligent UAV-based monitoring and mapping system for recording the weed distribution in wheat fields (weed-AI-seek)

Jing Liu[1], Pendar Alirezazadeh[2], Tom Kaufmann[3], Michael Schirrmann[2], Ludwig Schrenk[3], Frieder Stolzenburg[1]

1 Hochschule Harz, Friedrichstr. 57-59, 38855 Wernigerode, Deutschland
2 Leibniz-Institut für Agrartechnik und Bioökonomie e.V., Max-Eyth-Allee 100, 14469 Potsdam, Deutschland
3 CiS GmbH, Hansestraße 21, 18182 Bentwisch, Deutschland

## Abstract

Artificial intelligent systems have a great potential for identifying objects in unstructured data such as images. Unmanned aerial vehicles (UAV) open up new perspectives for weed monitoring because it enables very high-resolution remote sensing imagery (VHR) from which weed plants can be outlined even on the species-level. In this paper, a UAV platform is combined with an AI system (WeedAI) for wide-area real-time weed monitoring and mapping by implementing an improved YOLOv4 algorithm on an NVIDIA AGX Xavier board. In the current stage of our work, we have completed the integration of hardware and development of the software framework for the WeedAI system. The system is capable of automatically capturing VHR imagery at fixed intervals while flying at different flight altitudes. The improved YOLOv4 neural network has been developed to detect weeds with high accuracy and low latency. The deployment of the improved YOLOv4 neural network in the NVIDI AGX Xavier board is in progress. Further optimization is also planned to enable the WeedAI system to monitor a wider area in a single flight and provide the improved YOLOv4 neural network with the ability to differentiate between weed species.

## 1. Introduction

Because weeds are not uniformly distributed but occur in patches across the field with a spatially varying composition of weed species [1], spraying species-specific herbicides only at specific locations with weed occurrence can strongly reduce herbicide use and associated environmental impacts. This requires precise information about the site- and species-specific weed occurrence in an agricultural field. This paper proposes an intelligent, real-time, UAV-based weed monitoring and mapping system, called WeedAI system, to generate site- and species-specific distribution information for weeds in agricultural fields. For this purpose, very high-resolution image data (VHR) is captured at low altitudes and classified directly on the UAV platform using a weed recognition mode based on an improved YOLOv4 real-time object-detection system [2]. Recognition results are transmitted to the ground station in real time and converted into distribution information of weeds. This article will introduce the design process and software/hardware components of the WeedAI system from three aspects: system integration, improved YOLOv4 neural network development, and software framework. Finally, the preliminary results of the project will be showcased.

## 2. System Integration

As shown in Figure 1, the WeedAI system consists of three main parts: an onboard computer system, the UAV platform, and the ground station. Additionally, a Sony Alpha A6000 camera is mounted on a gimbal located underneath the drone and an Intel Realsense D455 depth camera is mounted on the main frame of the drone using a 3D-printed connector. They are connected to an NVIDIA Jetson board via USB 2.0 cables.

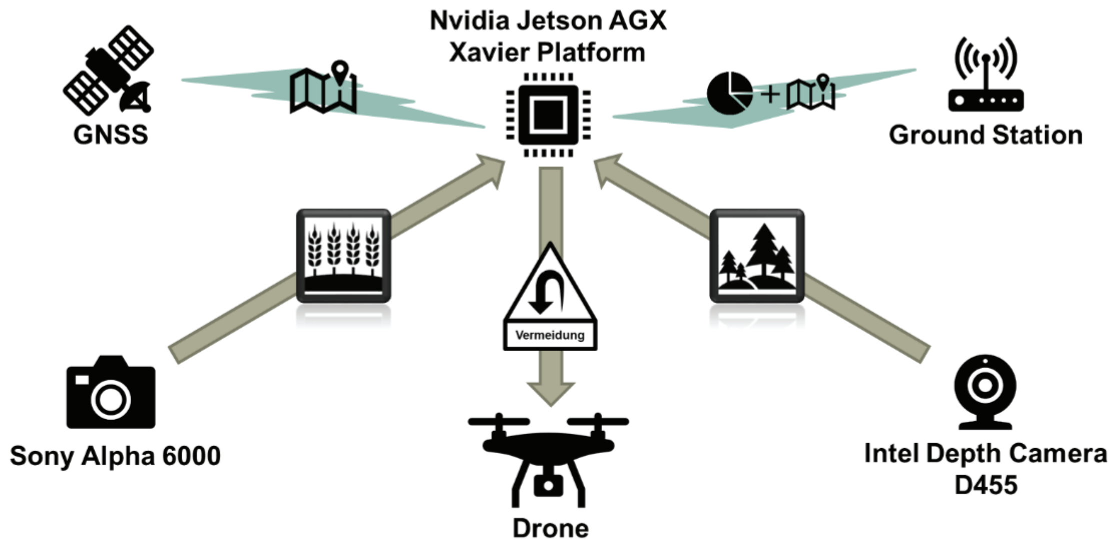


*Figure 1: System structure diagram.*

The onboard computer consists of an Jetson AGX Xavier 32 GB module board (NVIDIA corporation, Santa Clara, USA) with up to 32 TeraOps[1] of AI performance [3] and a carrier board X221 [4] (Auvidea GmbH, Denklingen, Germany). It is mounted on the top of the flying platform and connected via a UART serial interface to the drone's flight controller. The power is supplied through a 12V step-down switching power regulator that is directly connected to the primary battery bus (4S, nominal 14.8V).

The flying platform is an octorotor OktopusXL (CiS GmbH, Bentwisch, Germany), which features an 8 propellers design, uses the Pixhawk autopilot hardware and is equipped with an HD video transmission and control system. The default maximum lifting capacity is approximately 3kg. It uses a modified version of the open source Firmware PX4, which is optimized in regards of control, drivers and state estimation. Sensorwise, the OktopusXL features two accelerometers/gyroscopes, a barometer, Ublox M8N GPS and a laser rangefinder.

In order to overcome the change of the center of gravity and additional drag caused by the NVIDIA board (top-mounted), the flight control of the drone is optimized by using a wind and external drag-force estimation algorithm. The laser rangefinder data is added into the extended Kalman-Filter model to increase long term accuracy of the relative height estimation. General PID parameters were also tuned very carefully.

[1] Floating point operations per second

PX4 uses MAVLINK [5] in order to communicate with internal and external nodes. It is used for communication with the onboard computer as well as interaction with the MissionControl Software [6] (CiS GmbH, Bentwisch, Germany) to create and execute flight plans. The MissionControl Software was optimized to plan low-altitude flights with capabilities to follow terrain, adjust for uncertainties in elevation and monitor the in-flight status.

Visual Data from the Nvidia board can be shared via the HD video transmission and control system. The computer is connected via the HDMI protocol, and all data (including control, video and drone telemetry) is streamed via 2.4 GHz to a remote control unit. The technically possible range is approximately 30km on line of sight.

A Windows laptop is used as a ground station and a two-way full-duplex communication is established via a 433MHz telemetry radio connected to the NVIDIA board. This enables the ground station to transmit control commands and receive weed monitoring information within a range of 300m.

## 3. Improved YOLOv4 neural network design and training

In order to reduce the requirements for hardware configuration and to improve the performance of the recognition, an improved YOLOv4 deep neural network (cf. Figure 2) was designed and trained using the PyTorch framework [7].

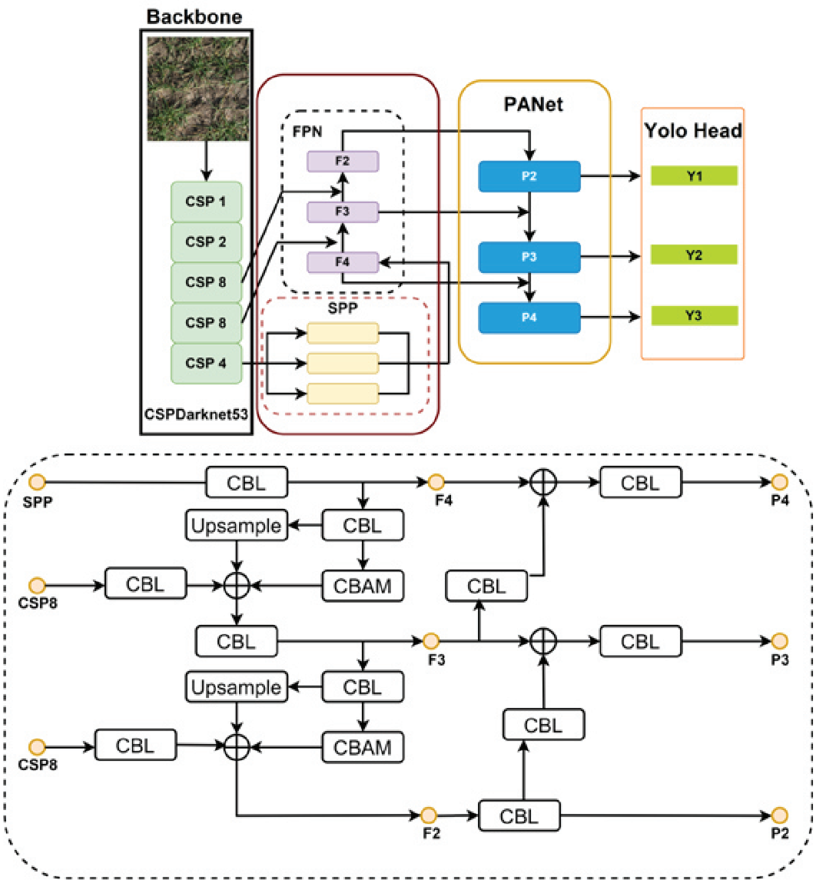


*Figure 2: Improved-YOLOv4 weed detection model. CBL: convolution, batch normalization, and Leak-ReLU.*

The CSPDarknet53 backbone borrows the cross-stage partial (CSP) from CSPNet [8] and adds a CSP on each of the five residual blocks, which improves learning ability of convolutional neural networks (CNNs) and allows it to maintain high performance while reducing network weight [9]. Shallow features in the CSPDarknet53 contain more target location information, such as contours and textures, but less semantic information. On the other hand, deeper features contain richer semantic information, whereas the object location information is coarse.

YOLOv4's neck consists of spatial pyramid pooling (SPP) [2], the feature pyramid network (FPN) [10], and the feature aggregator network (PANet) [11]. PANet attempts to improve the process of instance segmentation by retaining spatial information, which aids in proper pixel localization for prediction. FPN can pass deep semantic information to shallow layers and fuse semantic and location information to help YOLOv4 to detect small objects and improve overall weed detection performance.

In addition, to overcome the problems of occlusion, small targets for detection (weed plants), and high similarity in the weed detection process, the Convolutional Block Attention Module (CBAM) [12,13,14] is integrated into FPN. The extracted information is fused in order to use low-level location information in high-level feature maps. To improve the representation power of feature maps, we apply CBAM to the output of different layers of FPN and aggregate it with the output of different layers.

The experimental framework used in this study was developed on a Linux 64-bit operating system, utilized the Python 3.9 programming environment along with the PyTorch 1.11 deep learning framework for training and testing the model. Models were executed on a single NVIDIA GeForce RTX 2080Ti GPU with 11 GB of video memory and Cuda 11.2. The dataset was divided into training, validation, and test sets with a ratio of 8:1:1, respectively. In order to help annotators and increase the number of images, each image was divided into 6 parts to obtain 1662 images. The models were trained for 200 epochs using a batch size of 16. The training model was optimized using stochastic gradient descent (SGD) algorithm, with the momentum and learning rate values set at 0.95 and 0.001, respectively.

Further optimization is performed by the TensorRT[2] optimizer on the NVIDIA board, which performs hardware-specific optimization and finally generates a TensorRT inference engine optimized for the NVIDIA AGX Xavier module. The deployment of a TensorRT inference engine used the TensorRT runtime API with Python bindings.

## 4. Software framework development

The software framework of the WeedAI system is shown in Figure 3. Two data transfer programs run on the ground station and in the NVIDIA board separately to realize two-way communication. To save power, the NVIDIA board only calls the weed recognition and obstacle avoidance programs when it receives the command, and it can also be shut down remotely to save energy for flight-critical systems in case of low voltage.

[2] see https://developer.nvidia.com/tensorrt

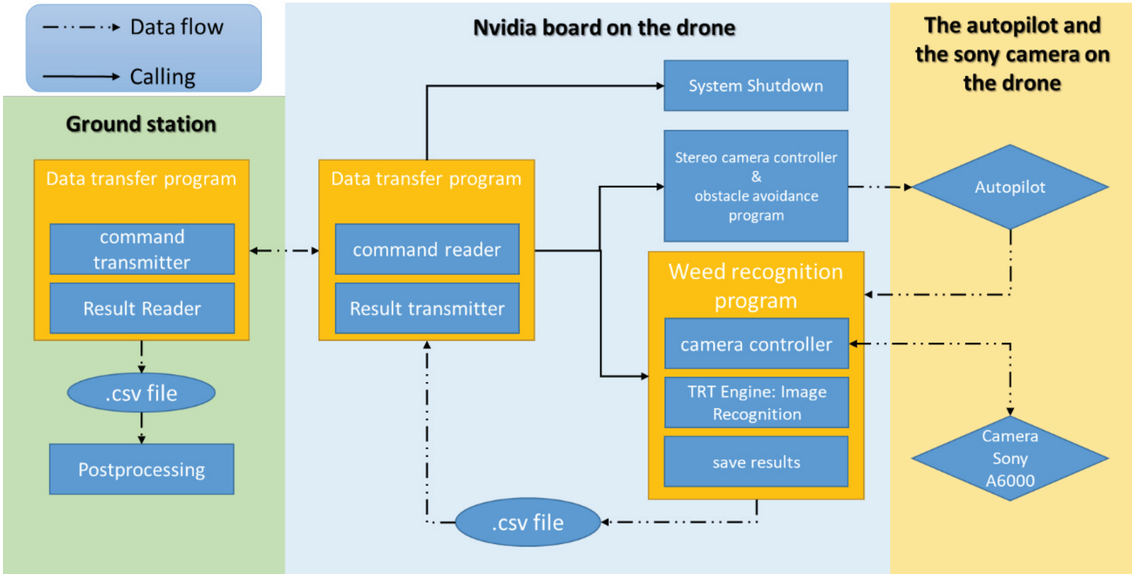


*Figure 3: Software framework diagram. Solid lines represent program calls, dashed lines represent data flow, and arrows indicate the direction of call and data flow.*

The weed recognition program controls the camera to take photos at fixed time intervals using the gphoto2 library,[3] obtaining the shooting location of the photo by reading the real-time flight information from the autopilot. The captured photo is immediately processed by an optimized inference engine running on the GPU to generate the recognition result, including the type, quantity and location of the weeds in the photo. This result together with shooting time and shooting location generates the weed monitoring information and is stored in a .csv file. At the same time, the on board data transfer program sends the newly added weed monitoring information to the ground station.

## 5. Results

The prototype of the WeedAI system was first tested on an experimental winter wheat field (ATB, Potsdam) to test the flight stability of the CiS drone at low altitude and the shooting performance of the camera. The weed recognition program was not tested in these test flights. All three flights were performed with different flight altitudes, namely 3m, 2m and 1.5m. The photos were taken by the Sony camera A6000 using a 50mm lens in fully automatic mode and with a fixed time interval of 1.5s. The number of photos and the flight duration are shown in Table 1.

| Flight altitude | Flight duration | Number of photos |
|---|---|---|
| 3m | 16min | 357 |
| 2m | 14min | 299 |
| 1.5m | 13min | 258 |

*Table 1: The number of photos and the flight duration per flight.*

The test result shows that the NVIDIA board is capable of controlling the Sony camera to capture photos. However, because the camera is always moving, it takes longer to focus than on the ground, therefore the capturing cannot always be completed within 1.5s. To overcome this issue, a Sony Multi-port connector is used to replace the regular

[3] see http://gphoto.org/

micro USB connector. The Sony Multi-port connector has 10 additional pinouts compared to the regular micro USB. The usage of each pinout is listed in Table 2. By shorting Pin 5 to Pin 2 and setting the focus mode of the camera to continuous autofocus, the camera can maintain focus continuously. In the further test we found this solution can effectively reduce the focus time for each shot. The downside is that the camera's power consumption increases significantly.

| Pin Num. | Usage |
|---|---|
| 1 | Power On / Off: Short to ground to toggle between Power On and Power Off |
| 2 | Ground |
| 3 | Composite Video Out |
| 4 | Shutter Release: Short to ground to trigger the shutter release |
| 5 | Focus: Short to ground to trigger focusing |
| 6 | Select: Select multiport functionalities / protocol |
| 7 | UART_RX |
| 8 | UART_TX |
| 9 | Reset |
| 10 | 2.8 Volt to 3.3 Volt output |

*Table 2: Pinout usages of the Sony Multi-port connector.*

For subsequent flights, the Sony camera will be controlled by the NVIDIA board to capture in manual expose mode with a shutter time of 1/800s and an aperture of F5.6, to reduce blurring caused by vibration and relative motion. By the test flight, it was also found that the CiS drone can fly smoothly at low altitudes, but the terrain tracking ability is closely related to the accuracy of the terrain information on the electronic map.

The improved YOLOv4 neural network was tested in labor on the test set with 167 annotated images. The algorithm loss function metric is IoU (intersection over union). If the score for each class exceeds the threshold, namely 0.5, the score is sorted. The score and prediction box position are then subjected to non-maximal suppression (NMS) processing. Finally, the prediction result is the bounding box with the highest probability. The precision-recall curve is shown in Figure 4. The mean average precision to detect weeds was 65.68%. The improved model achieved precision, recall, and F1-score, 93.04%, 60.81%, and 57.00%, respectively. It should be noted that this result is a preliminary result obtained by the model trained on a small training set, as the annotation work has not yet been completed. As shown in Figure 5, the improved YOLOv4 model could optimize the detection by enhancing useful features (weed) in the modeling process and overcome occlusion and noise in the field environment. It improved the ability to detect weeds even with very small sizes. Furthermore, the proposed model could detect weeds in a dense and nested environment, which is one of the most significant challenges in weed detection (Figure 5 (b) and (c)).

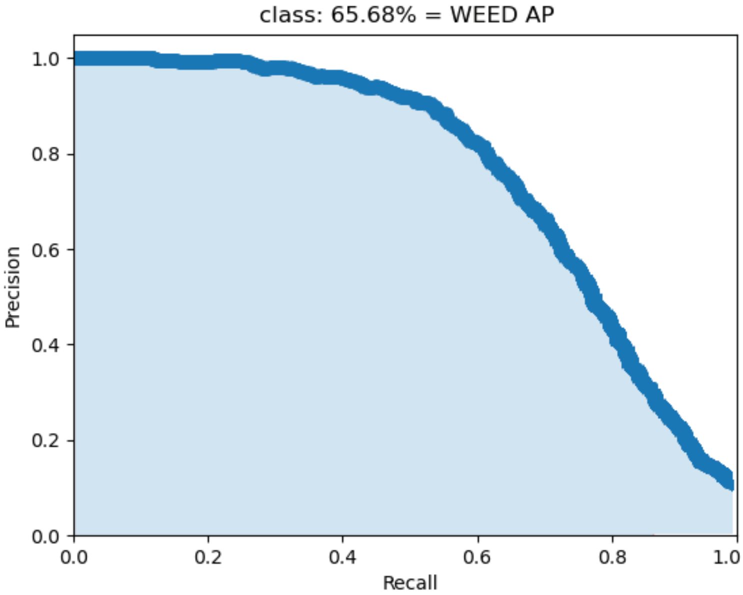


*Figure 4: Precision–recall curves of the detection results of the model with test set.*

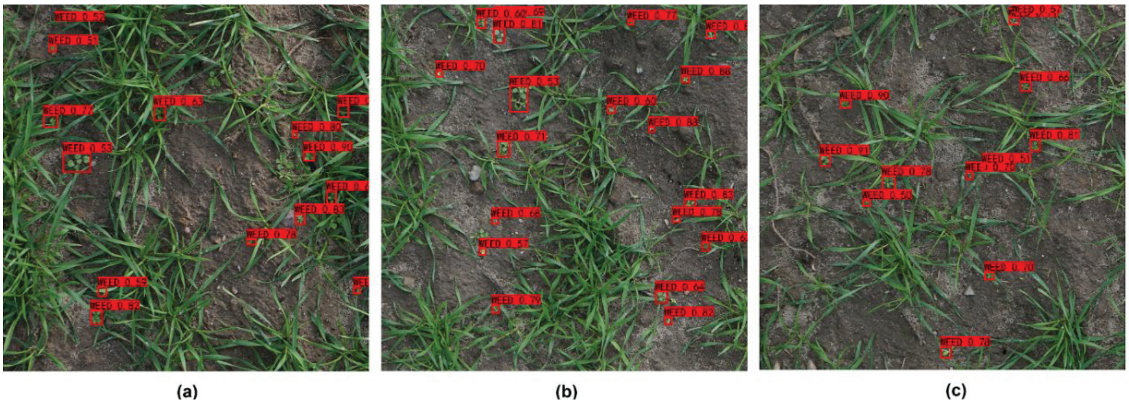


*Figure 5: Examples of weed detection results of test set with different challenges. (a) occlusion; (b) dense; (c) nested.*

## 6. Conclusion and outlook

The prototype of the WeedAI system has been developed and tested on both hardware and software parts. The flight performance meets the requirements of the high-payload and low-altitude flight. The on-board computer was integrated in the system and uses the developed software framework to realize the control of the system and the transmission of data. The improved YOLOv4 neural network exhibits robust small object detection capability. Currently, the network deployment is in progress, and we aim to compare different deployment methods, such as ONNX Runtime and TensorRT Runtime, to obtain meaningful results. Hardware optimizations will focus on reducing the interval between image capture and optimizing terrain-following flight capability. Further training and structure optimization are necessary to enhance the improved YOLOv4 network's ability to differentiate between weed species.

## Literature


1. Christensen, S., Heisel, T., Walter, A. M., & Graglia, E. (2003). A decision algorithm for patch spraying. Weed Research, 43(4), 276-284.

2. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Corr abs/2004.10934. https://arxiv.org/abs/2004.10934

3. NVIDIA Jetson Modules, Support, Ecosystem, and Lineup. (2022, 1. November). NVIDIA Developer. https://developer.nvidia.com/embedded/jetson-modules

4. AUVIDEA X221, -LC, -AI Technical reference manual. (Jun.2022). Version 1.6 https://auvidea.eu/download/manual/X221/X221_Manual

5. Mavlink Messaging. MAVLink Messaging | PX4 User Guide, https://docs.px4.io/main/en/middleware/mavlink.html

6. Kaufmann, T. (2022). Software für Drohnen. https://www.cis-rostock.de/kopter/software/

7. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., and others (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). https://dl.acm.org/doi/10.5555/3454287.3455008

8. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391). https://doi.org/10.1109/CVPRW50498.2020.00203

9. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916. https://link.springer.com/chapter/10.1007/978-3-319-10578-9_23

10. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125). https://ieeexplore.ieee.org/document/8099589

11. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768). https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8579011

12. Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19). https://link.springer.com/chapter/10.1007/978-3-030-01234-2_1

13. Alirezazadeh, P., Rahimi-Ajdadi, F., Abbaspour-Gilandeh, Y., Landwehr, N., & Tavakoli, H. (2021). Improved digital image-based assessment of soil aggregate size by applying convolutional neural networks. Computers and Electronics in Agriculture, 191, 106499. https://dl.acm.org/doi/abs/10.1016/j.compag.2021.106499

14. Alirezazadeh, P., Schirrmann, M., & Stolzenburg, F. (2022). Improving Deep Learning-based Plant Disease Classification with Attention Mechanism. Gesunde Pflanzen, 1-11. https://doi.org/10.1007/s10343-022-00796-y