



AI-Autonomous Robots for Agriculture – Weeding with Laser



Bild von [Chaiyan Anuwatmongkolchai](#) auf [Pixabay](#)



Funded by the Horizon 2020 programme
of the European Union



UNIVERSITY OF
COPENHAGEN



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Applications of AI in Agriculture

Crop Yield prediction & Price forecasts

Identify the output yield of crops and forecast prices for the next few weeks will help the farmer to obtain maximum profit

Intelligent spraying

AI sensors can detect weed affected areas and can precisely spray herbicides in the right region reducing the usage of herbicides

Predictive Insights

Insights on "Right time to sow the seeds" for maximum productivity. Insights on the impacts created by the weather conditions



Artificial Intelligence in Agriculture



Agriculture Robots

Using Autonomous robots for harvesting huge volumes of crop at a higher volume and faster pace

Crop and soil monitoring

Using ML/AI, we can monitor the crop health for diagnosing pests/soil defects, nutrient deficiencies in soil, etc.

Disease Diagnosis

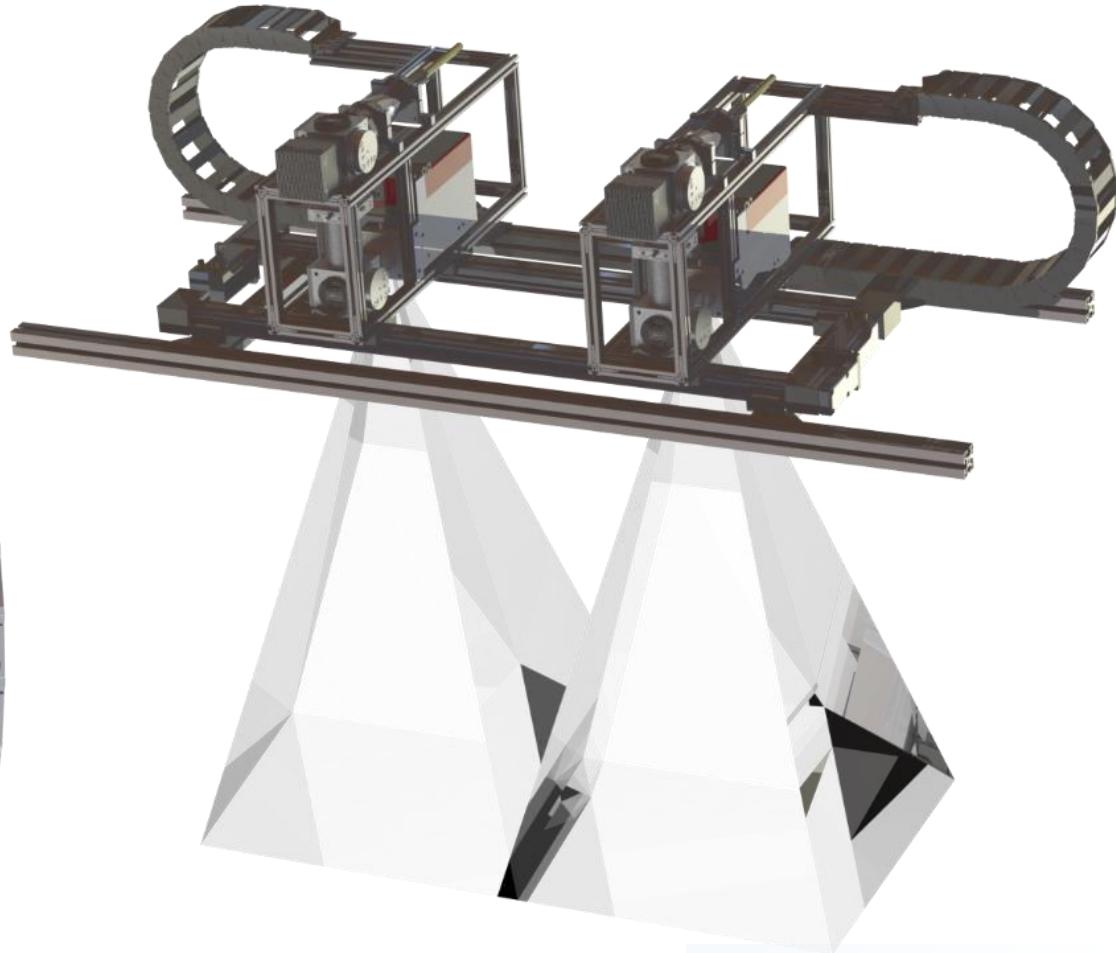
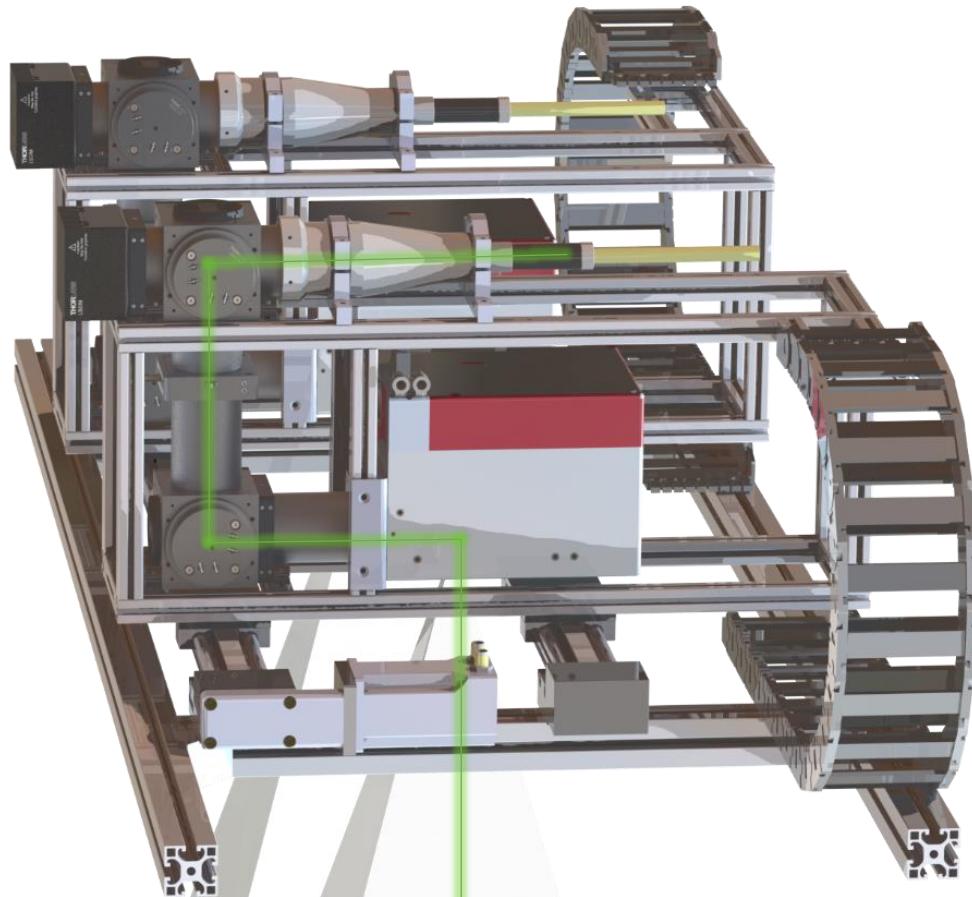
Prior information and classification of Plant diseases help farmers control the disease through proper strategy.

Source: www.wipro.com/holmes/towards-future-farming-how-artificial-intelligence-is-transforming-the-agriculture-industry/

- WeLASER aims to merge current technologies to build and push close to the market a precision weeding system based on high power laser sources and autonomous vehicles
- main objective:
 - of eliminating the use of herbicides and their health/environmental adverse effects
 - Simultaneously improving productivity and competitiveness



- Robot with Implement
- Weeding Scanner



WeLASER – System breakdown

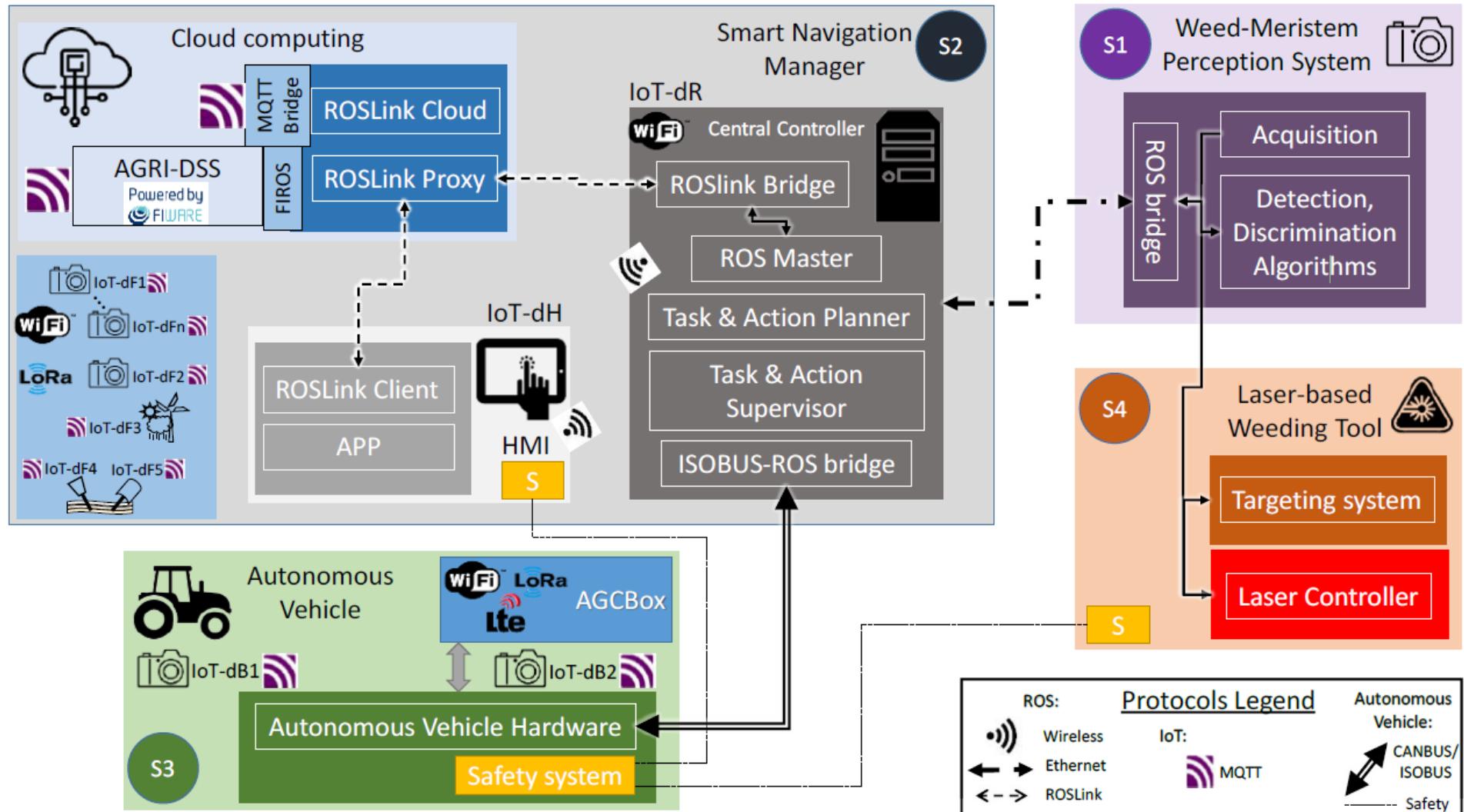


Fig. 2.13. WeLASER system architecture: communication interfaces and protocols

WeLASER – System breakdown

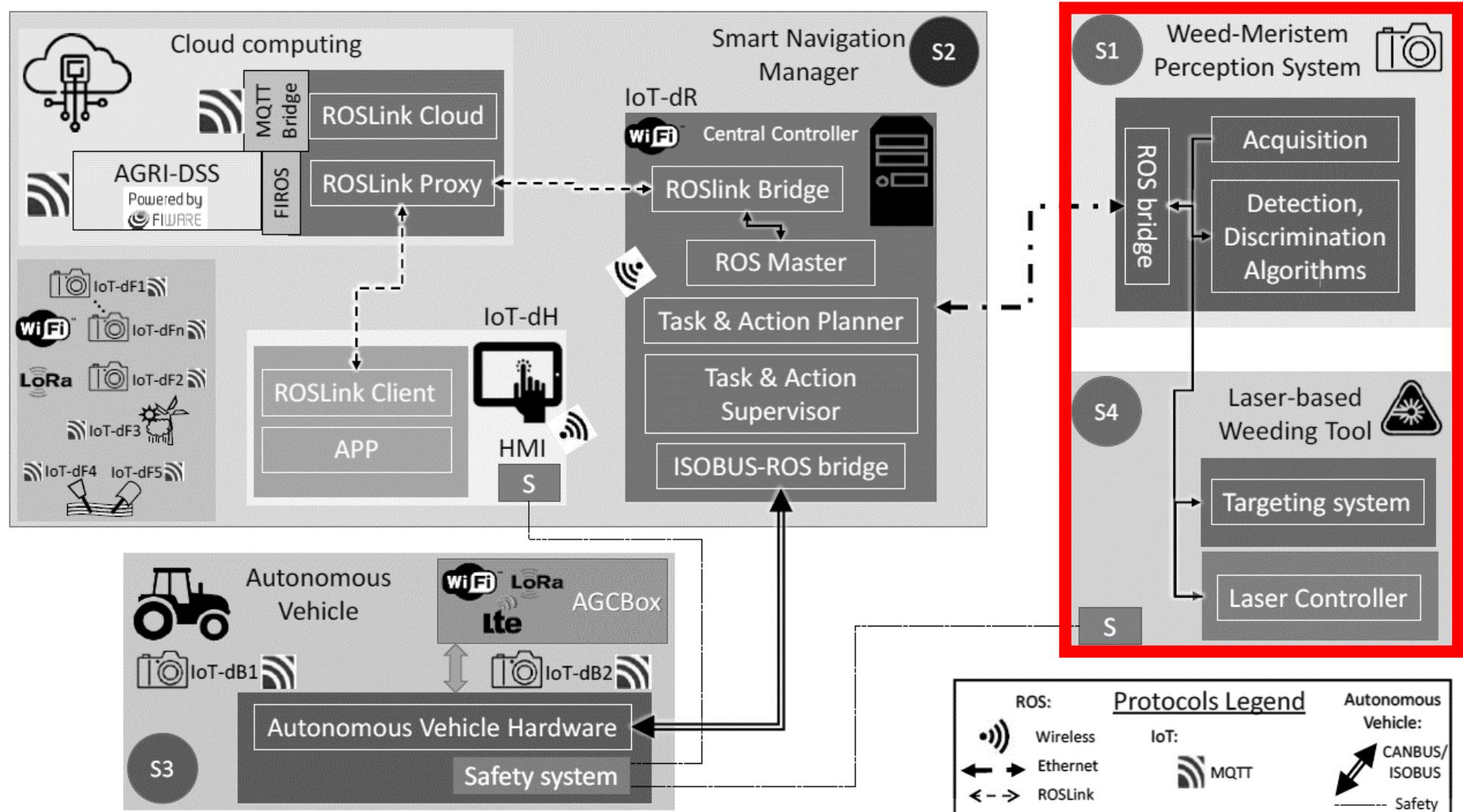


Fig. 2.13. WeLASER system architecture: communication interfaces and protocols

Laser treatment of juvenile cornflowers





- **Laser weeding process**

- Plant recognition
- Classification as „crop“ vs. „weed“
- Find impact point (apical meristem, stem)
- Target position to laser targeting system
- Track target
- Apply (lethal) dose
- Go to next target



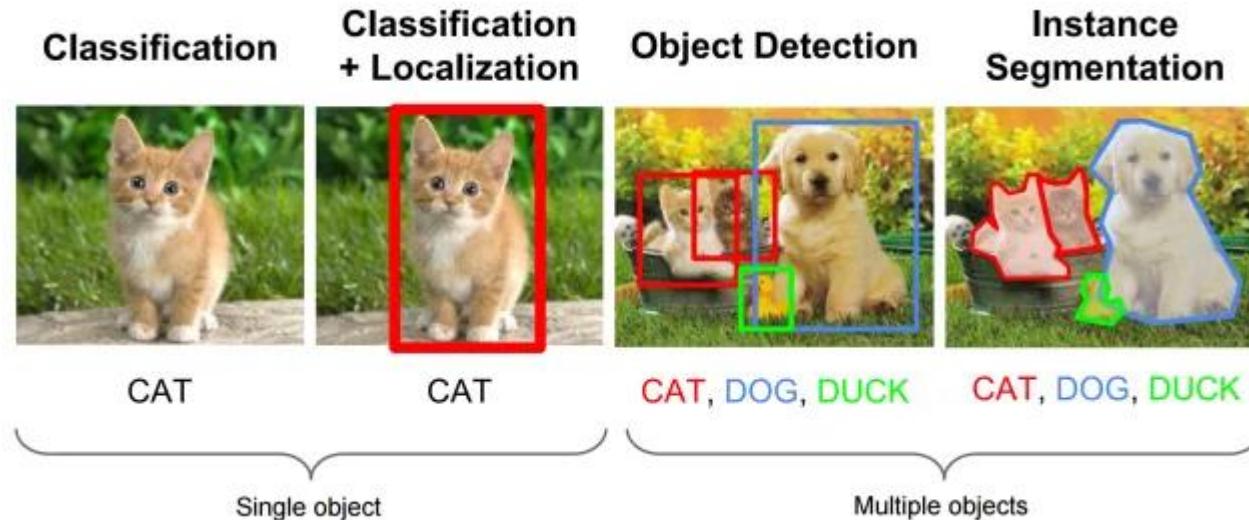
- **Laser weeding process**

- Plant recognition
- Identify weeds to treat
- Find impact point (apical meristem, stem)
- Target position to laser targeting system
- Track target
- Apply (lethal) dose
- Go to next target

- **AI job?**

- Object detection
- Classification
- Object detection/classification
- -
- (Object tracking)
- -
- -





- **Object Localization**

A bounding box or enclosing region is determined in the tightest possible manner in order to locate the exact position of the object in the image.

- **Image Classification**

The localized object is fed to a classifier which labels the object.

- **Semantic Segmentation**

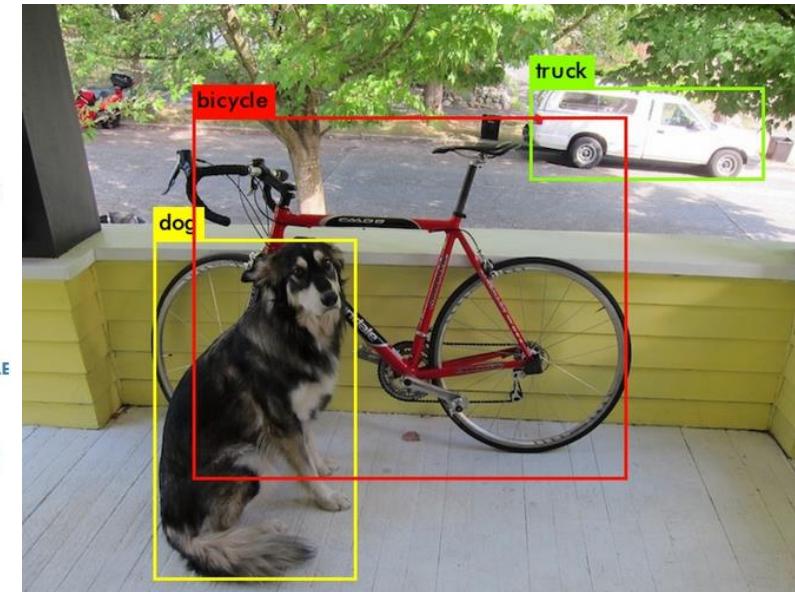
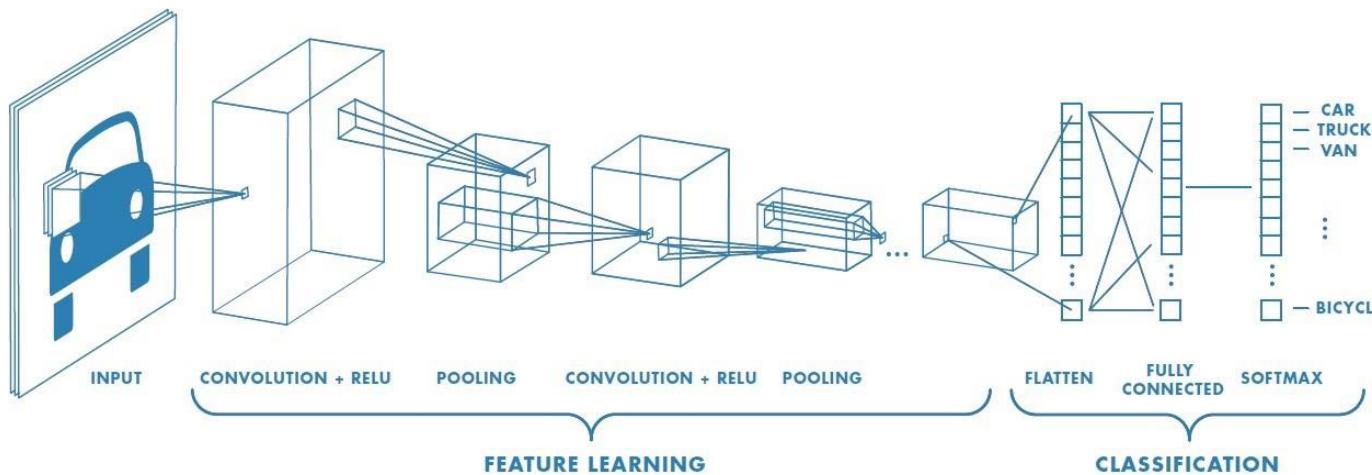
Each pixel in the given image is linked to a particular class label. These segments are then used to find the interactions / relations between various objects.

- **Instance Segmentation**

Associate a class label to each pixel. Similar to semantic segmentation, except that it treats multiple objects of the same class as individual objects / separate entities.

- **Perception System**

- Plant recognition with artificial intelligence and neural networks



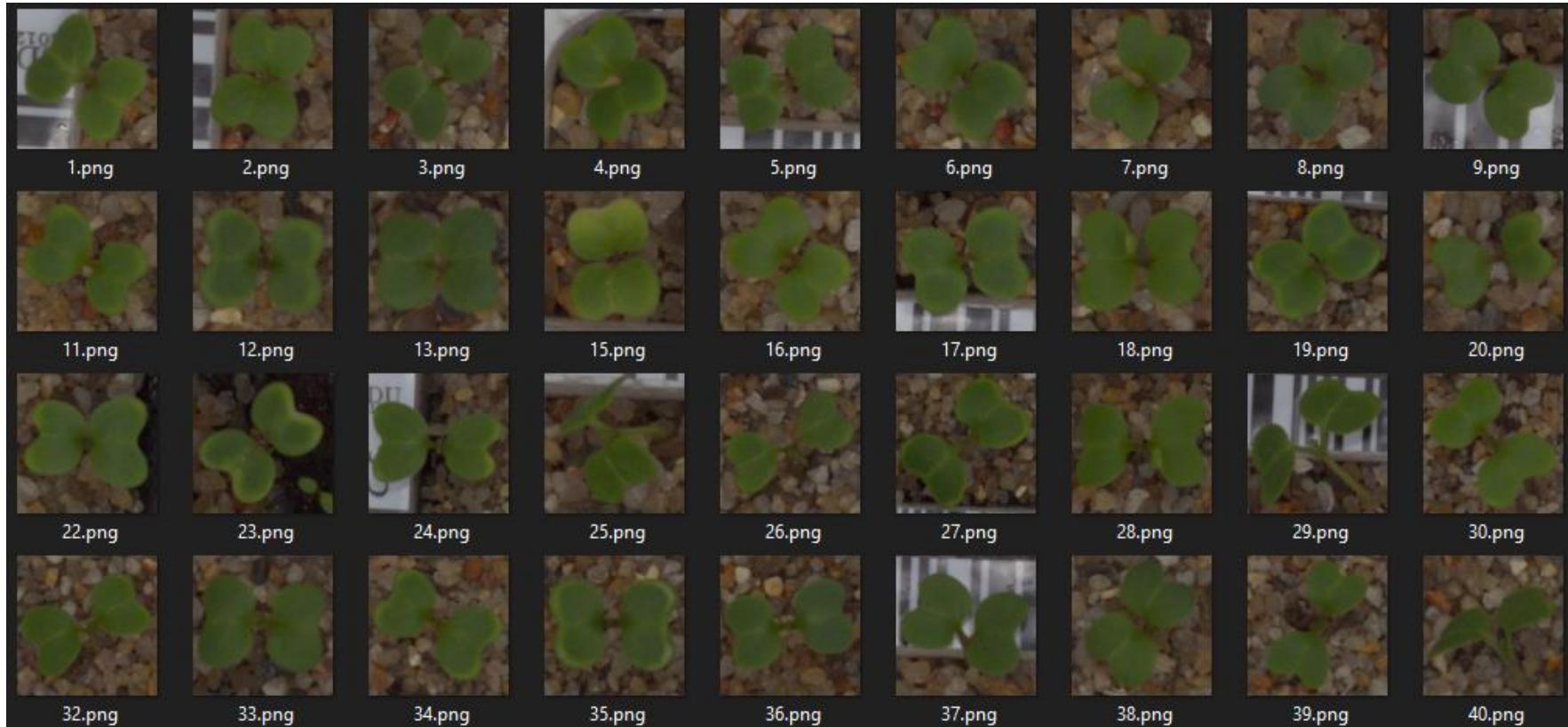
Bottleneck of agricultural AI application

- Availability of high-quality data for training
- Agricultural image datasets require efforts and costs for acquisition and pre-processing



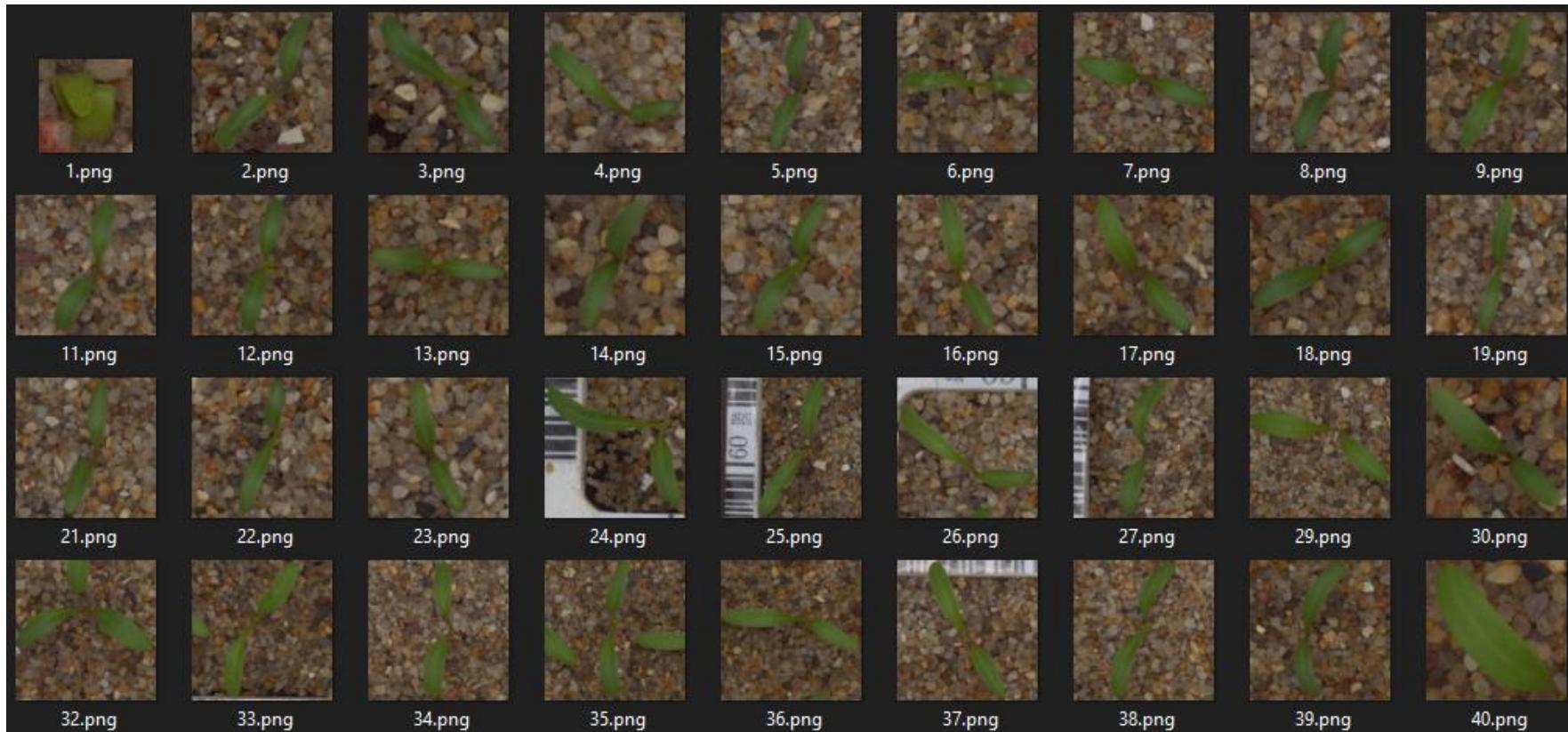
- **Perception System**

- Plant recognition with artificial intelligence and neural networks
- In addition to existing plant databases, new specific training data is acquired



- **Perception System**

- Plant recognition with artificial intelligence and neural networks
- In addition to existing plant databases, new specific training data is acquired



- **Perception System**

- Plant recognition with artificial intelligence and neural networks
- In addition to existing plant databases, new specific training data is acquired



- **Perception System**

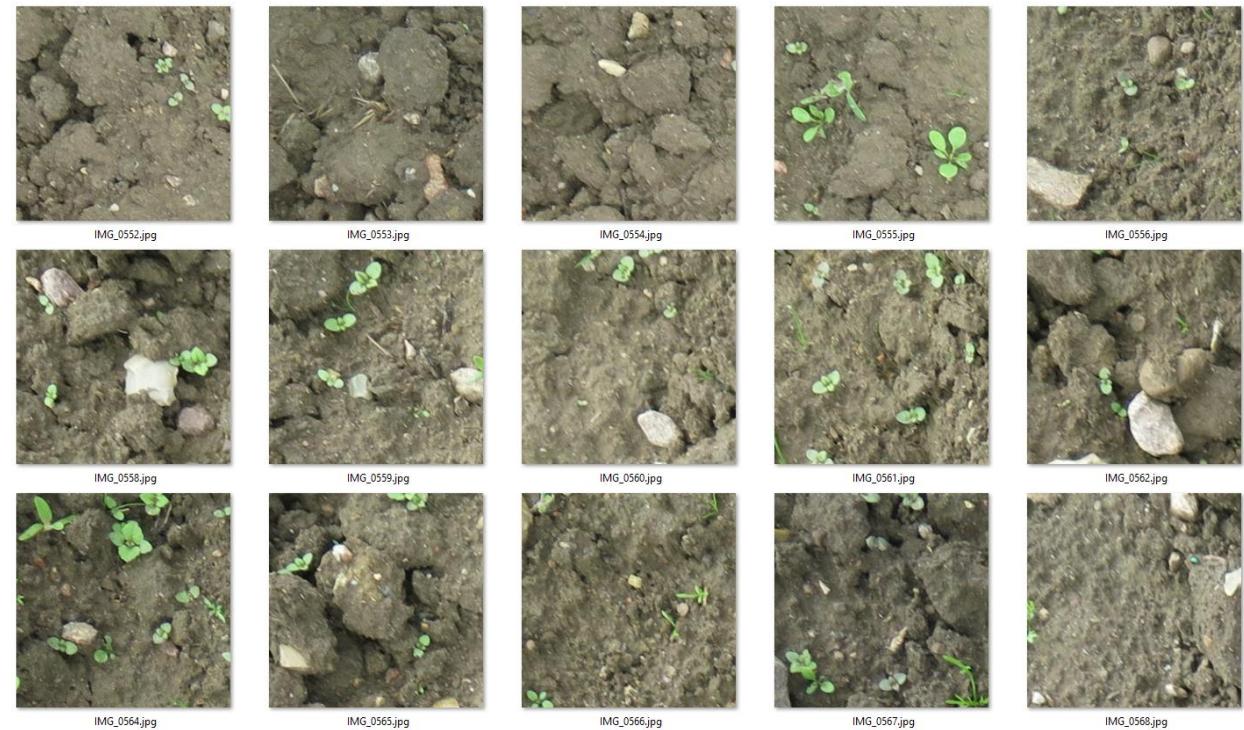
- Plant recognition with artificial intelligence and neural networks
- In addition to existing plant databases, new specific training data is acquired



- Own Dataset with labelled images has to be acquired
- How to get large labelled images?



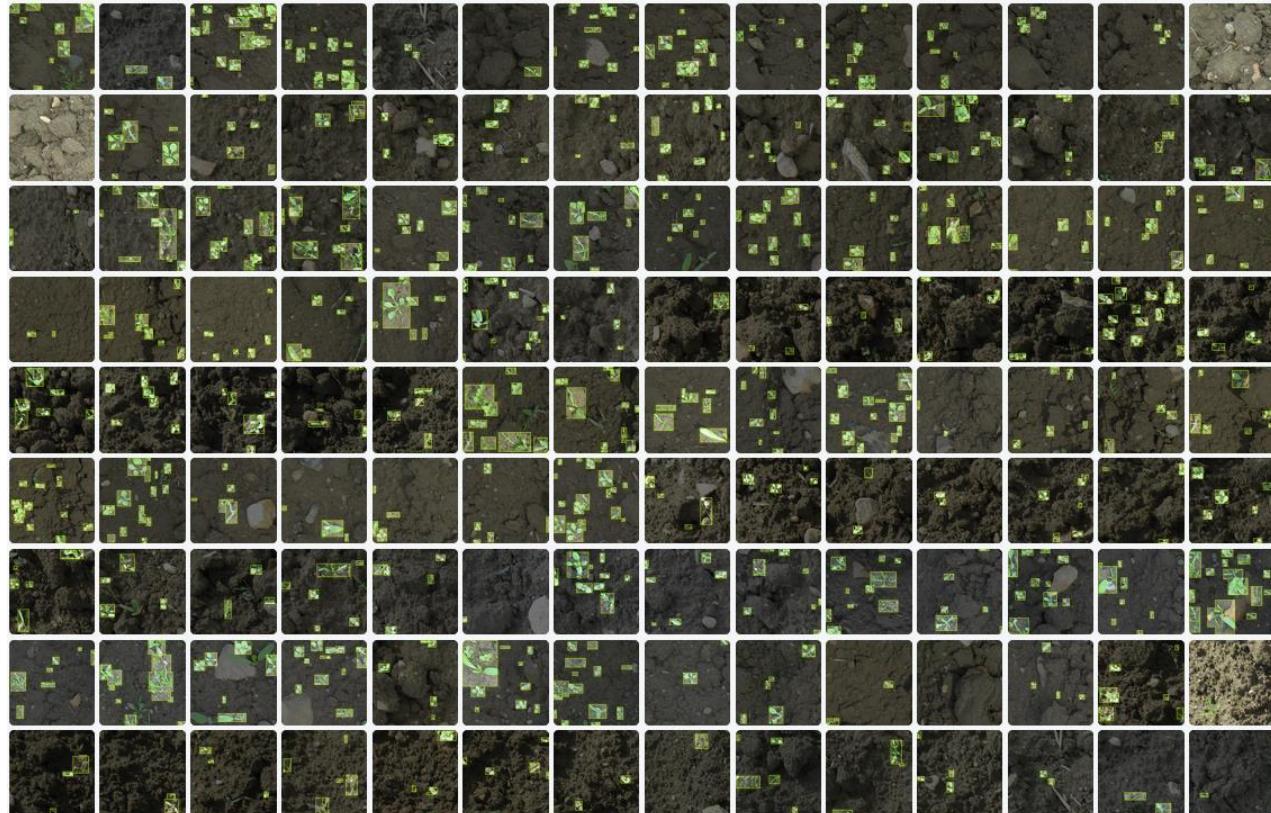
- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images



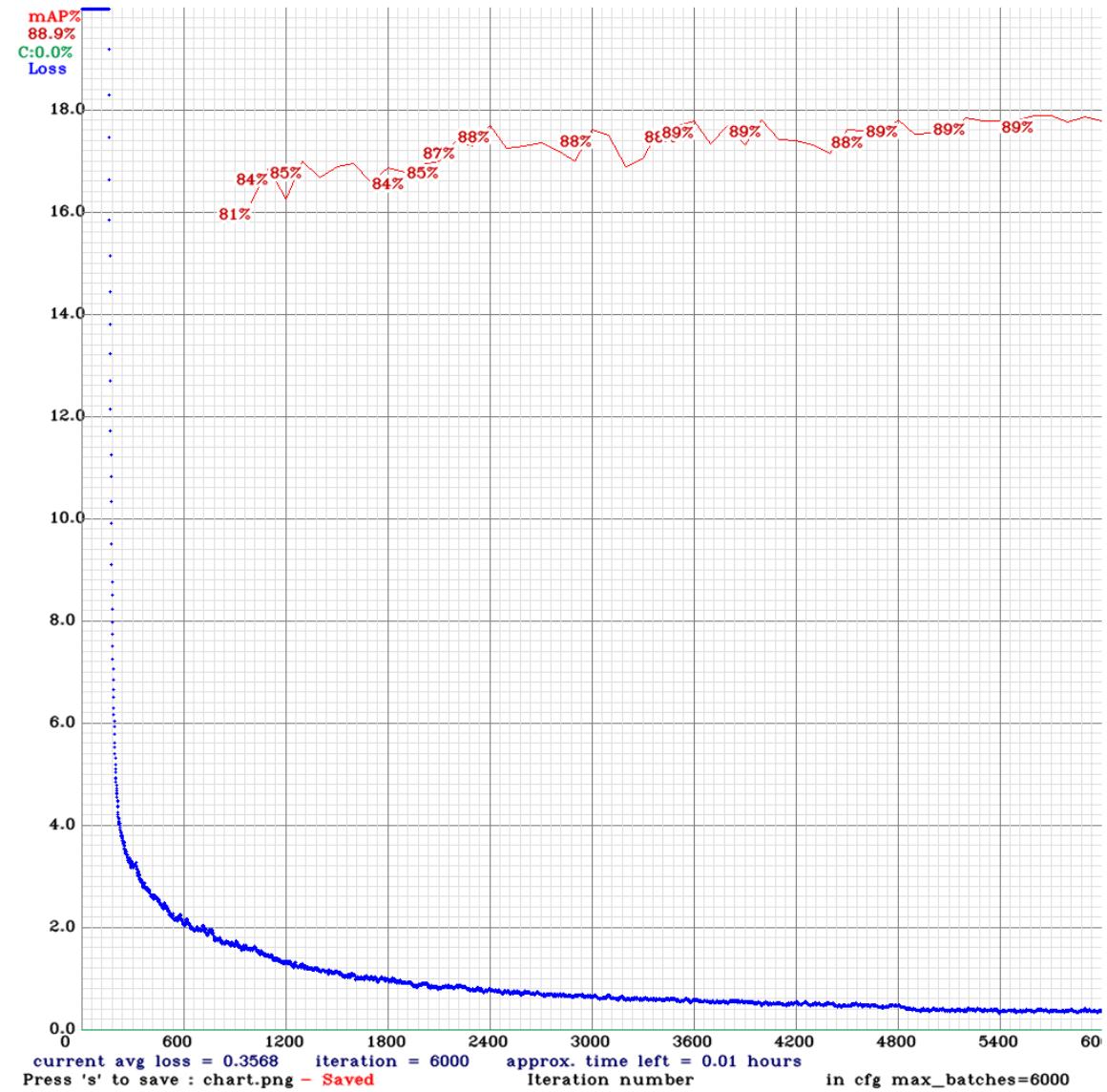
- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images
 - Label 416x416 px images



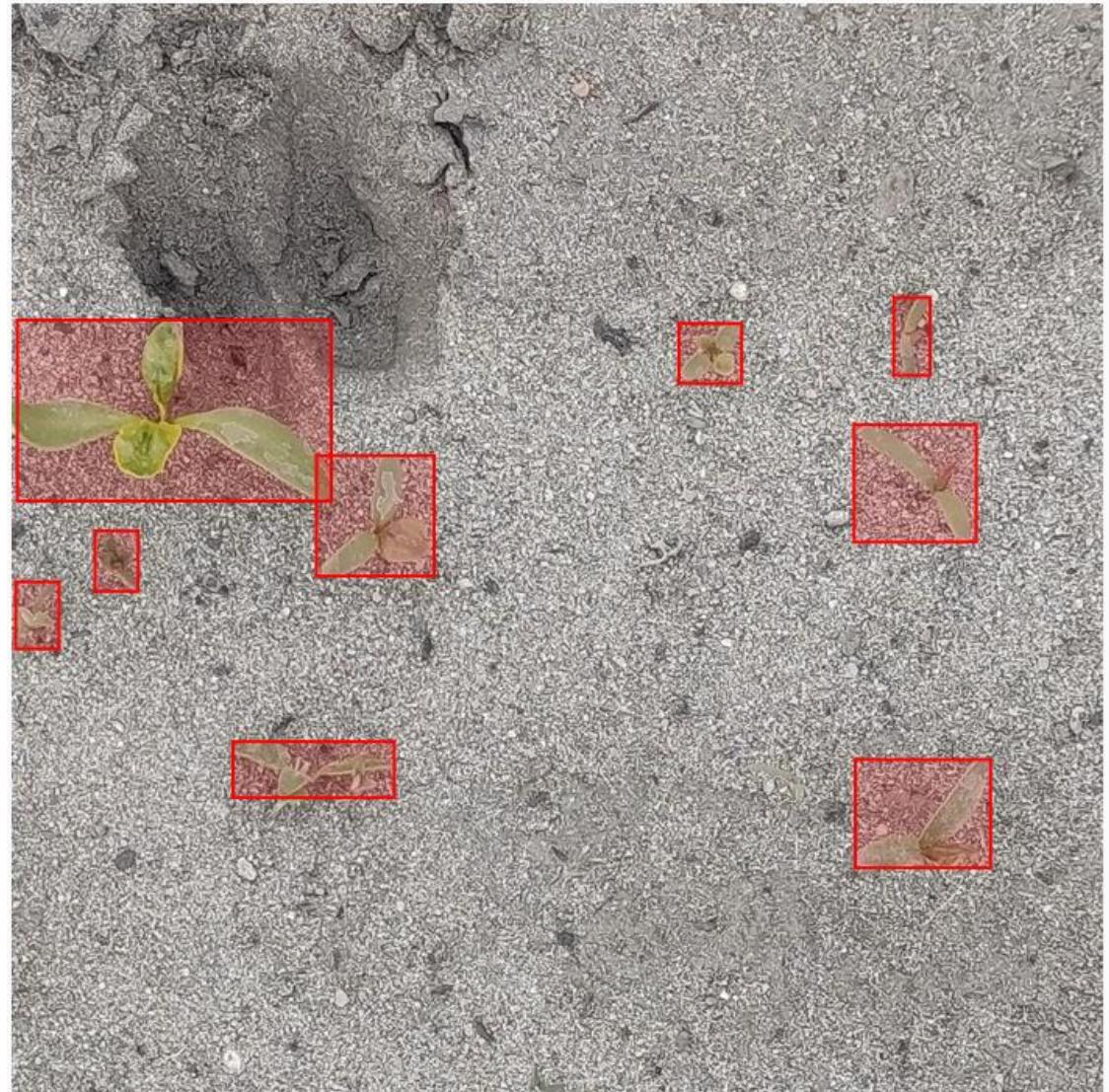
- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images
 - Label 416x416 px images



- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images
 - Label 416x416 px images
 - 3x Data Augmentation
 - Train full-scale 416 model



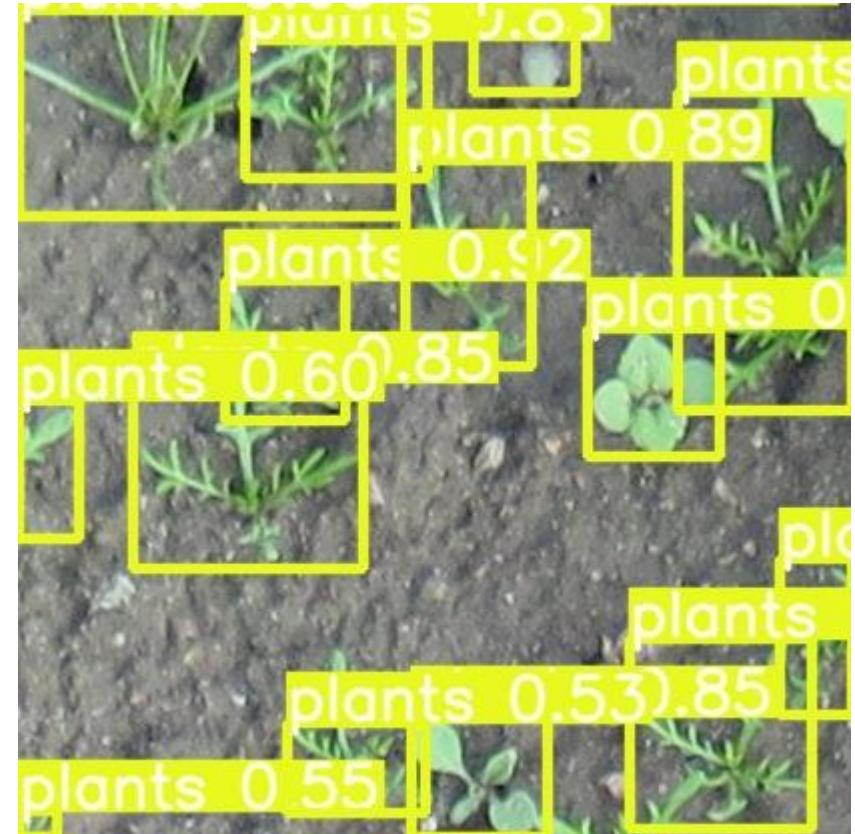
- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images
 - Label 416x416 px images
 - 3x Data Augmentation
 - Train full-scale 416 model
 - Apply on 1024x1024 images from German dataset
 - Improve labels by hand



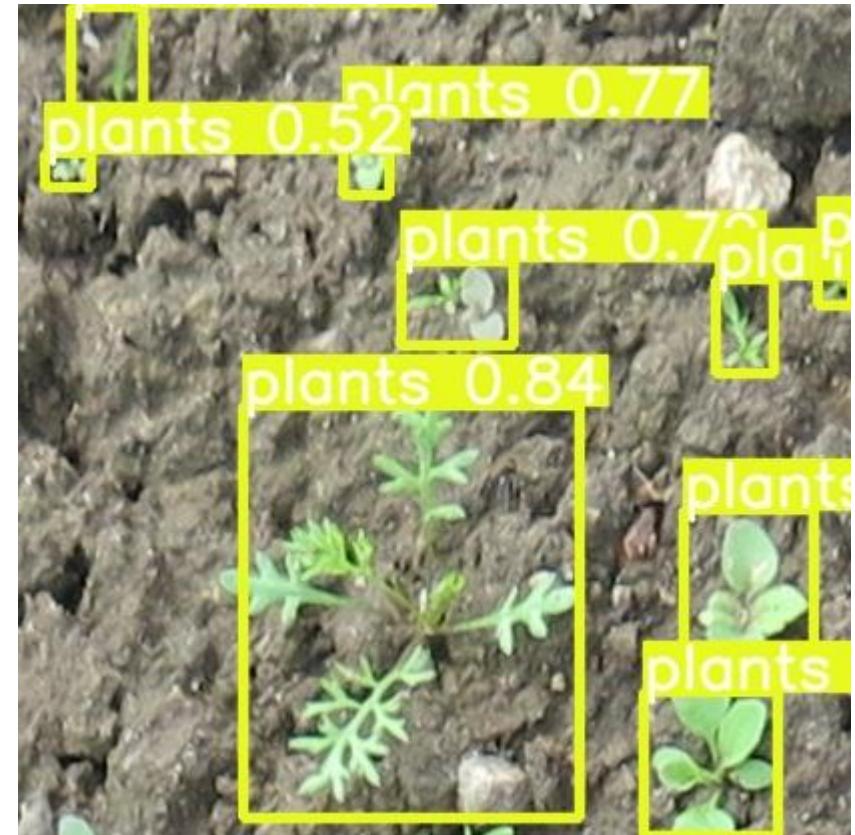
- Own Dataset with labelled images has to be acquired
- How to get large labelled images?
 - Split large unlabelled images
 - Label 416x416 px images
 - 3x Data Augmentation
 - Train full-scale 416 model
 - Apply on 1024x1024 images from German dataset
 - Improve labels by hand
 - Retrain with new data

```
c:\ Auswählen C:\Windows\System32\cmd.exe - darknet.exe detector train data/obj.data cfg/yolov4-tiny... — □ X
D:\yolov4\darknet>darknet.exe detector train data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny.conv.29 -dont_show -map
CUDA-version: 11040 (11040), cuDNN: 8.2.4, GPU count: 1
OpenCV version: 4.5.3
Prepare additional network for mAP calculation...
0 : compute_capability = 860, cudnn_half = 0, GPU: NVIDIA GeForce RTX 3090
net.optimized_memory = 0
mini_batch = 1, batch = 4, time_steps = 1, train = 0
layer filters size/strd(dil) input output
0 Create CUDA-stream - 0
Create cudnn-handle 0
conv 32 3 x 3/ 2 1024 x1024 x 3 -> 512 x 512 x 32 0.453 BF
1 conv 64 3 x 3/ 2 512 x 512 x 32 -> 256 x 256 x 64 2.416 BF
2 conv 64 3 x 3/ 1 256 x 256 x 64 -> 256 x 256 x 64 4.832 BF
3 route 2 1/2 -> 256 x 256 x 32
4 conv 32 3 x 3/ 1 256 x 256 x 32 -> 256 x 256 x 32 1.208 BF
5 conv 32 3 x 3/ 1 256 x 256 x 32 -> 256 x 256 x 32 1.208 BF
6 route 5 4 -> 256 x 256 x 64
7 conv 64 1 x 1/ 1 256 x 256 x 64 -> 256 x 256 x 64 0.537 BF
8 route 2 7 -> 256 x 256 x 128
9 max 2x 2/ 2 256 x 256 x 128 -> 128 x 128 x 128 0.008 BF
10 conv 128 3 x 3/ 1 128 x 128 x 128 -> 128 x 128 x 128 4.832 BF
11 route 10 1/2 -> 128 x 128 x 64
12 conv 64 3 x 3/ 1 128 x 128 x 64 -> 128 x 128 x 64 1.208 BF
13 conv 64 3 x 3/ 1 128 x 128 x 64 -> 128 x 128 x 64 1.208 BF
14 route 13 12 -> 128 x 128 x 128
15 conv 128 1 x 1/ 1 128 x 128 x 128 -> 128 x 128 x 128 0.537 BF
16 route 10 15 -> 128 x 128 x 256
17 max 2x 2/ 2 128 x 128 x 256 -> 64 x 64 x 256 0.004 BF
18 conv 256 3 x 3/ 1 64 x 64 x 256 -> 64 x 64 x 256 4.832 BF
```

- Test on unknown images from cropped DK dataset



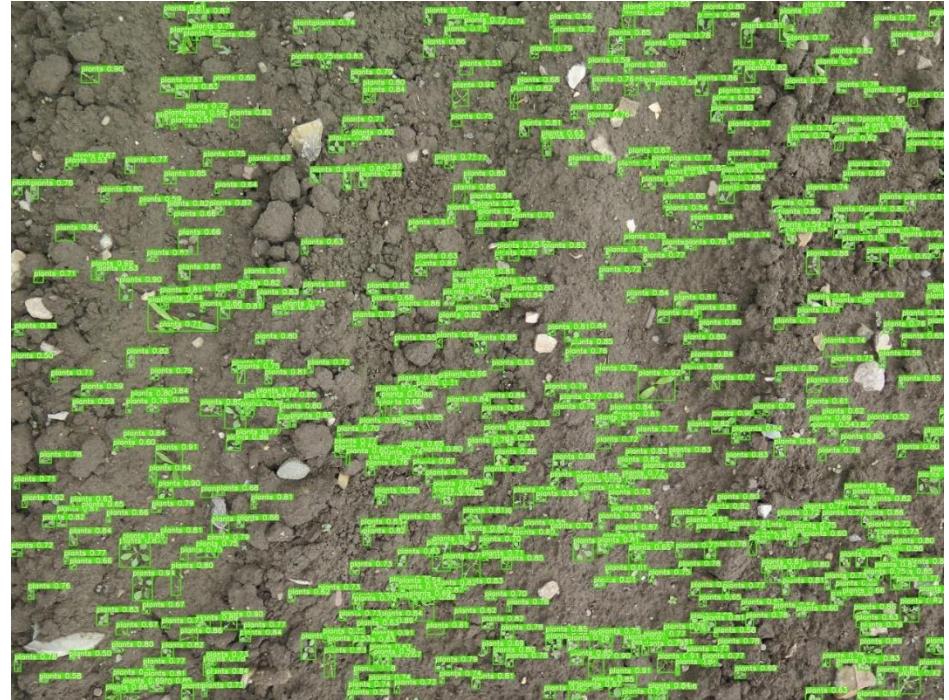
- Test on unknown images from cropped DK dataset



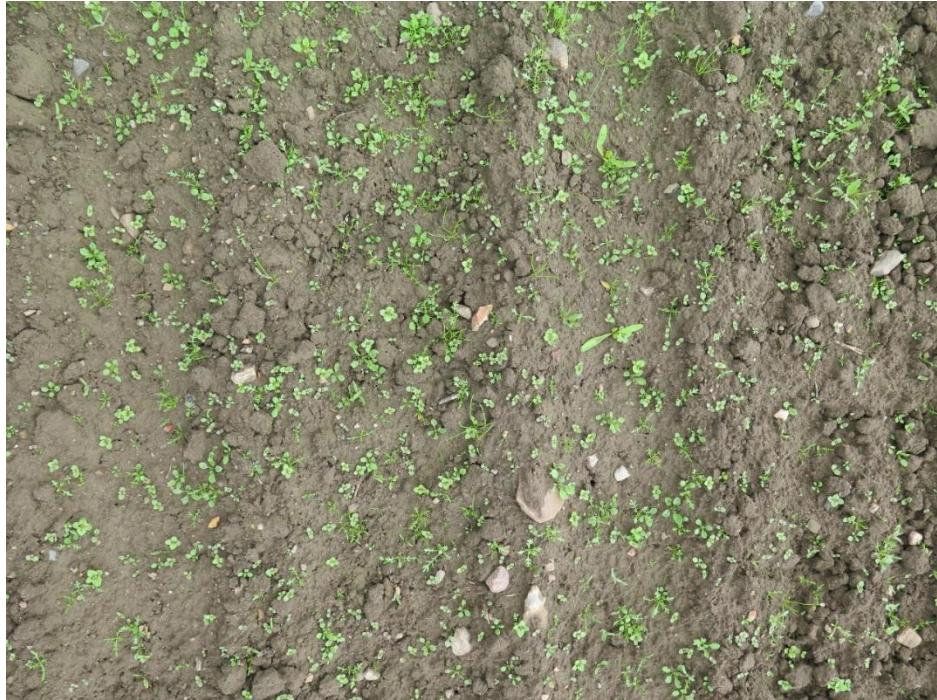
- Test on unknown images from cropped DK dataset



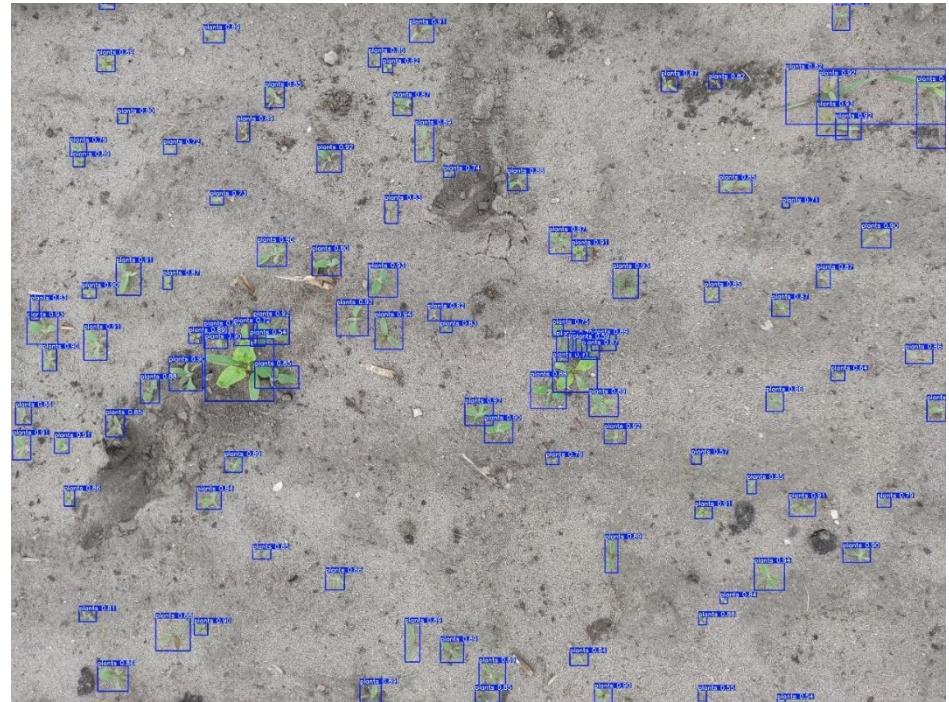
- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset



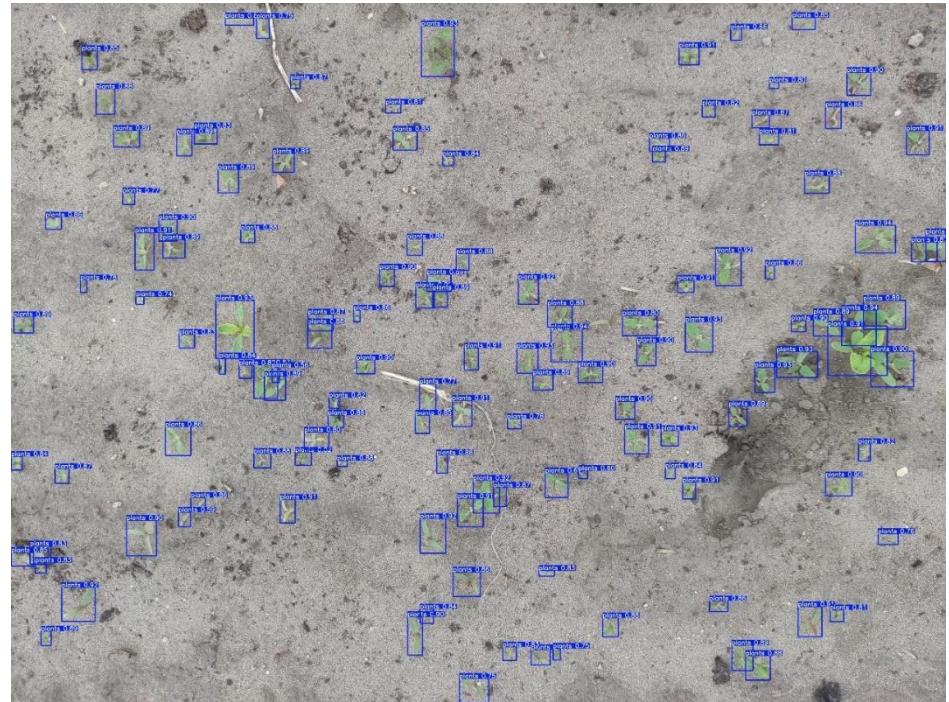
- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset



- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset
- Test on unknown images from DE dataset



- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset
- Test on unknown images from DE dataset



- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset
- Test on unknown images from DE dataset
- Test on unknown images from different field

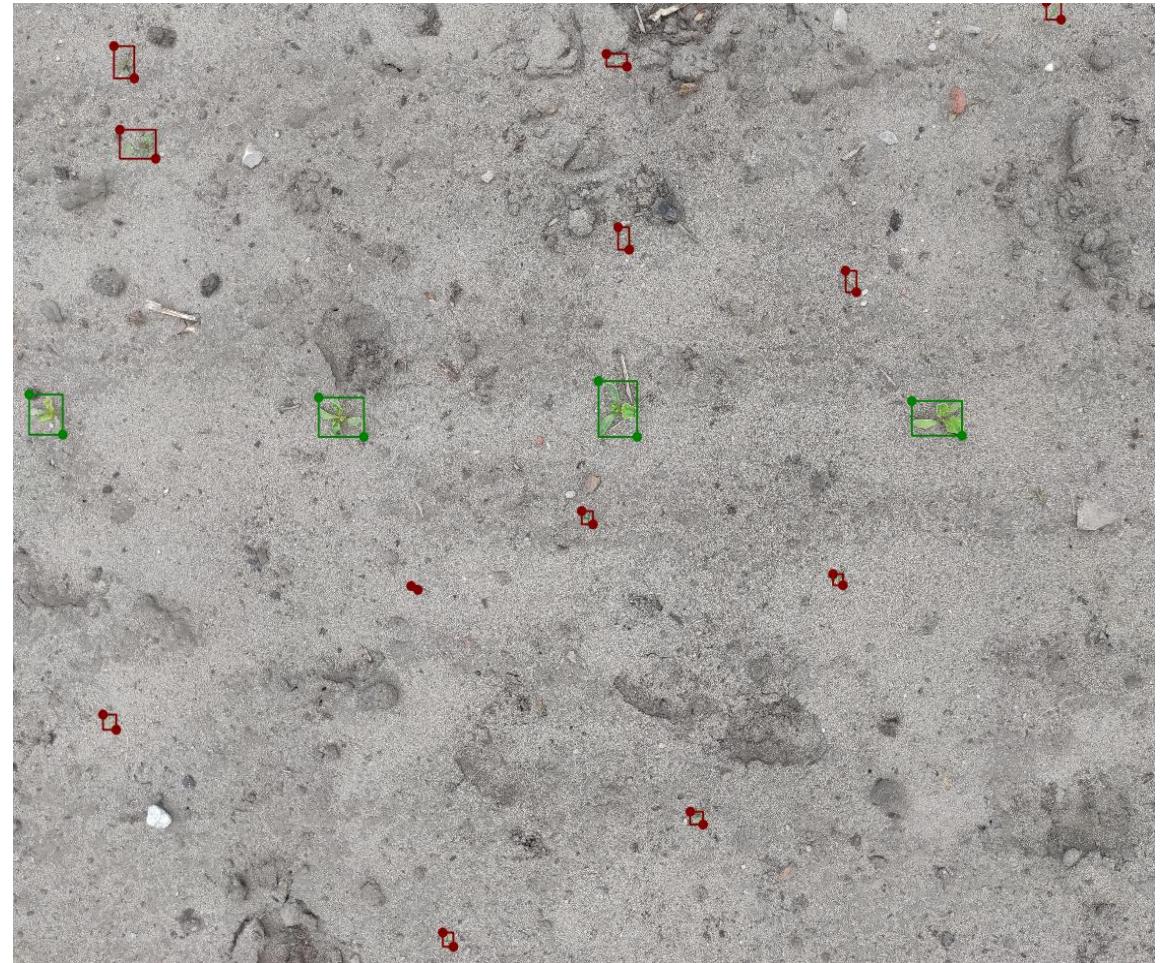


- Test on unknown images from cropped DK dataset
- Test on unknown images from full DK dataset
- Test on unknown images from DE dataset
- Test on unknown images from different field



- **Crop/Weeds Distinctions**

- Using YoloR CNN
- 30 ms per „Full HD“ image
- ~95% precision
- Flexible future class count



- **Meristem Detection**

- Top-Down Method:

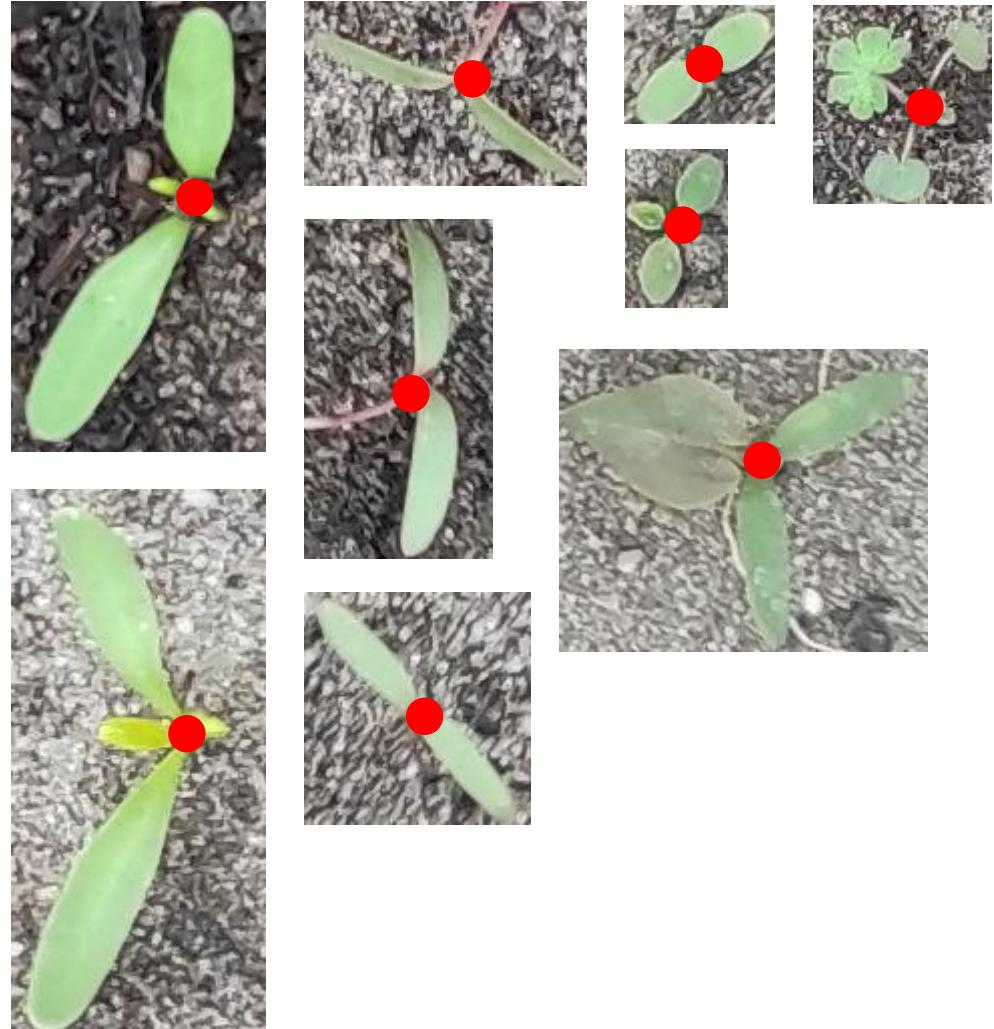
- ❖ Cut out plants from large picture



- **Meristem Detection**

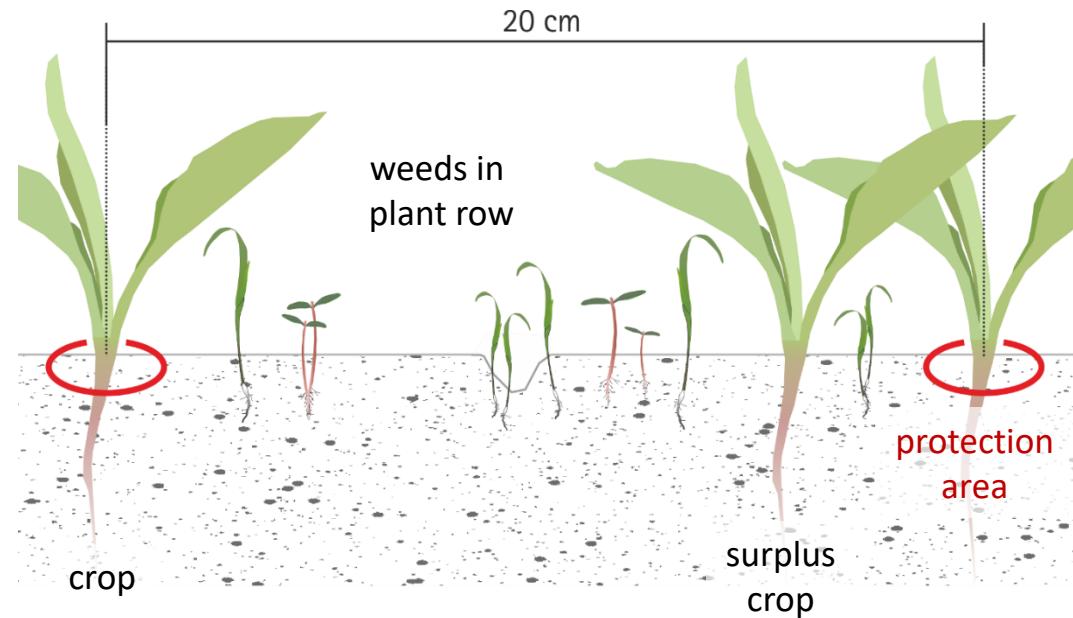
- Top-Down Method:

- ❖ Cut out plants from large picture
 - ❖ Small convolutional neural network to predict impact point
 - ❖ ~80% precision
 - ❖ Total computation time of 50 ms per large image



- **Characteristics**

- Precise single plant treatment
- AI target classification and localization
- Mechanism: thermal effect
- Efficient for young plants up to BBCH14
- Compatible to conservation soil cultivation
- Low-wear technology
- Precision + selectivity → enabling technology for weed management allowing for maximal biodiversity



Graphics: C. Marx LZH



AI-Autonomous Robots for Agriculture – Weeding with Laser



Thank you for your attention



Funded by the Horizon 2020 programme
of the European Union



UNIVERSITY OF
COPENHAGEN



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

