Multi-Pose Time-Series Tomato Database for Fine-Grained Categorization with Deep Learning

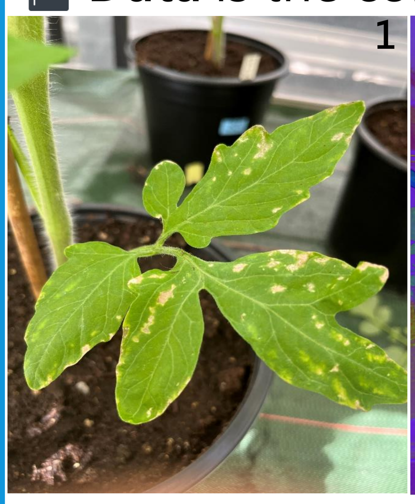
Julius Kühn-Institut
Federal Research Centre for Cultivated Plants

Yujie Zhang[1], Sven Reichardt[1]

[1]Julius Kühn-Institut, Federal Research Centre for Breeding Research on Horticultural Crops, Institute for Breeding Research on Horticultural Crops (ZG), Data Analytics Horteculturae (DaAnHort), Erwin-Baur-Str. 27, 06484, Quedlinburg, Germany.
yujie.zhang@julius-kuehn.de, sven.reichardt@julius-kuehn.de

Overview

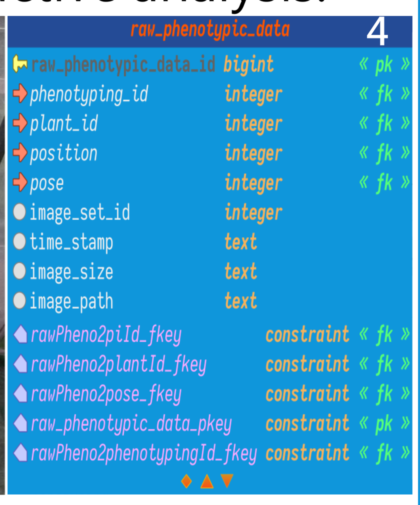
Plant phenotype detection plays a crucial role in assessing crop health, growth patterns, and identifying diseases.[1]
Conventional algorithms are time-consuming and require significant human effort, making them unsuitable for **large-scale phenotyping.**[2]
Deep learning techniques have shown great promise in automating this process by enabling accurate and efficient analysis of annotated plant image data.[3]
Data is the cornerstone of AI algorithms in predictive analysis.[4]



Abnormal Leaf

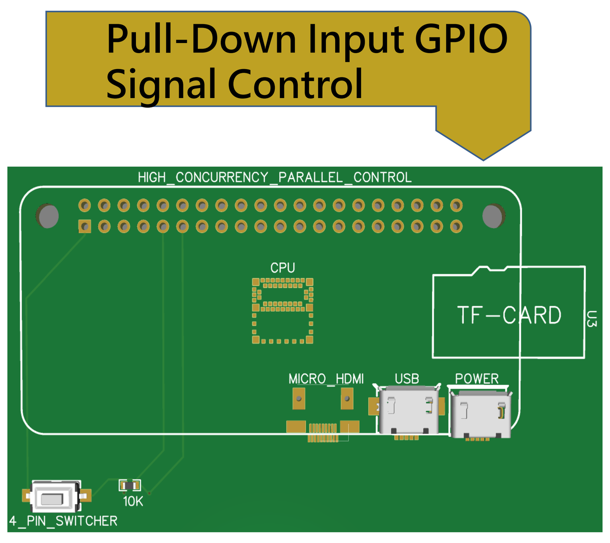
Classic Segmentation

Data Annotation

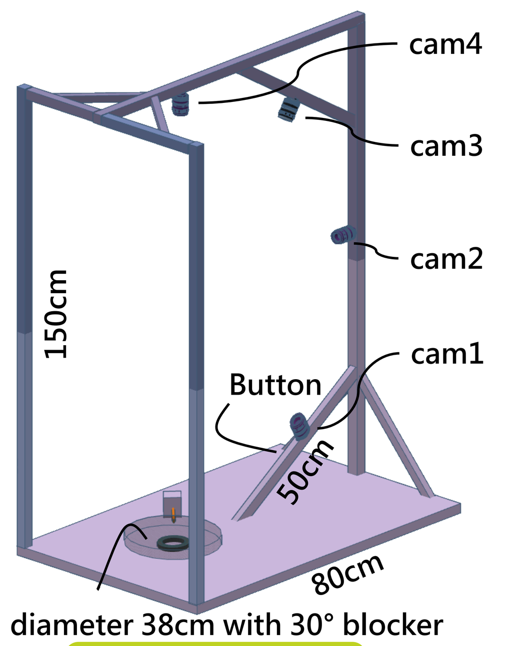


Database

Data Station



3D PCB module



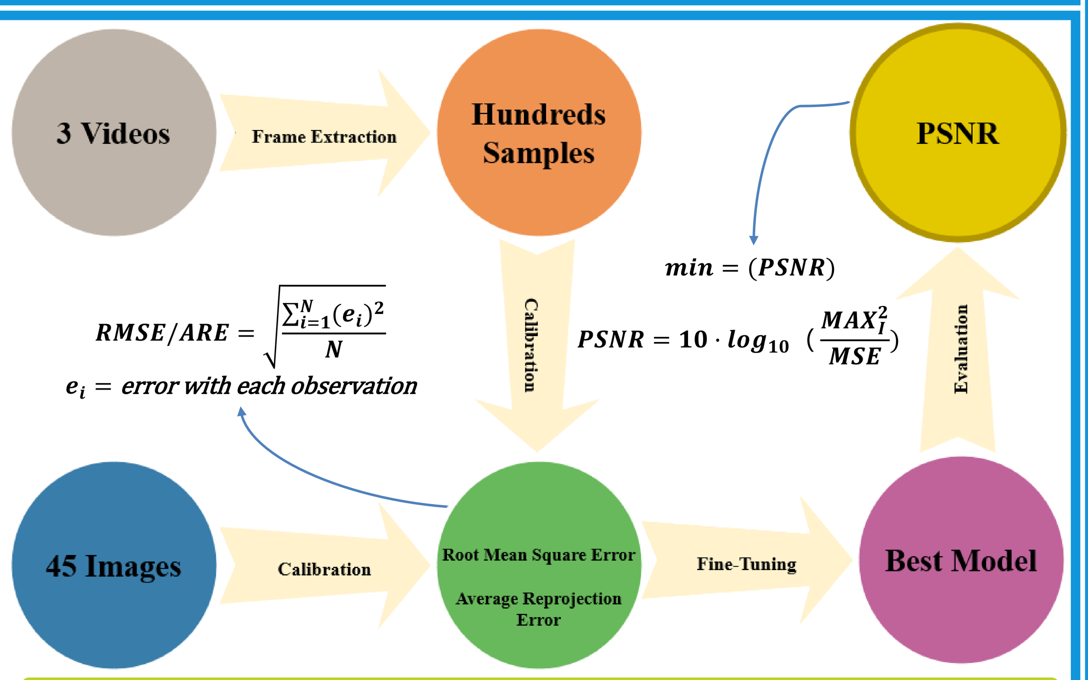
3D CAD module


Phenotyper v1.0

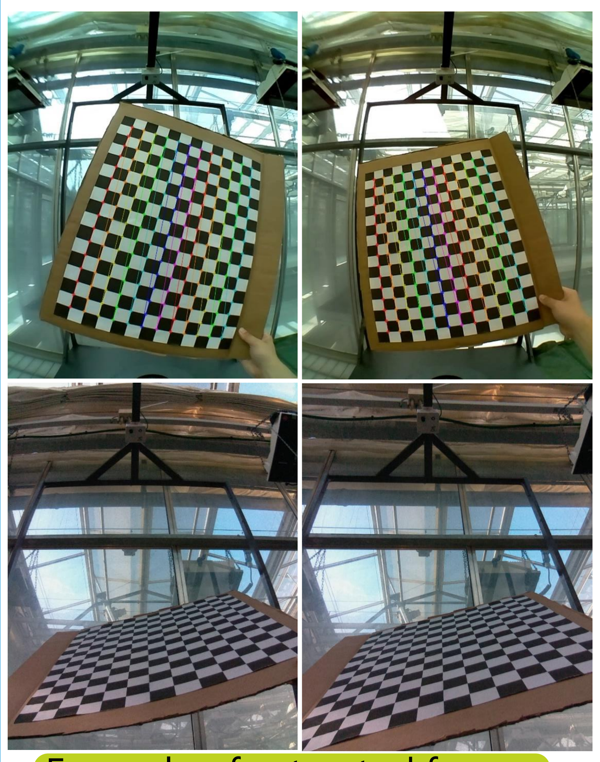
Lock mechanism is used in concurrent programming to enhance code robustness and stability.
Multiprocessing pool is employed to manage parallel computing and enhance resource utilization.
IP map based socket connection is used for remote control of cameras and the signal control unit.

Quality Control

The quality control of the dataset is primarily assessed using RMSE and ARE during the calibration phase, while the **average peak signal-to-noise ratio** (PSNR) is mainly used for overall quality control after calibration.



Fisheye and super wide angle camera calibration progress


Example of extracted frames and calibration results

Root Mean Square Error:

Method	Cam 1	Cam 2	Cam 3	Cam 4
Video1	0.0935	30.9048	0.0967	0.1395
Video2	0.1206	3.2342	0.1171	552.9743
Video3	0.1147	0.8835	5515.1196	10.0304
Image	0.0747	0.8950	0.0927	0.0728

Average Re-projection Error:

Method	Cam 1	Cam 2	Cam 3	Cam 4
Video1	0.0918	30.7810	0.0940	0.1320
Video2	0.1190	2.1214	0.1131	63.3698
Video3	0.1084	0.3053	477.4222	5.2166
Image	0.0732	59.2439	0.0829	0.0687

Peak signal-to-noise ratio (PSNR):

Method	Cam 1	Cam 2	Cam 3	Cam 4
Best Video	28.71	29.60	30.72	29.60
Image	28.69	30.97	30.54	30.10

Data Structure

Plant side shoots data has a **hierarchical structure.**
Maintaining and retrieving data with a **ltree structure** will be more cost-effective and convenient.
We use an **auto-incrementing primary** key, and image data are stored as **paths** in the database.
We present phenotyping data from different **growth cycles** of tomato plants. The data includes height, internodes, side shoots, and buds, flowers, fruits at each stage of the tomato plant through **time-series sampling** for **fine-grained** categorization purposes.

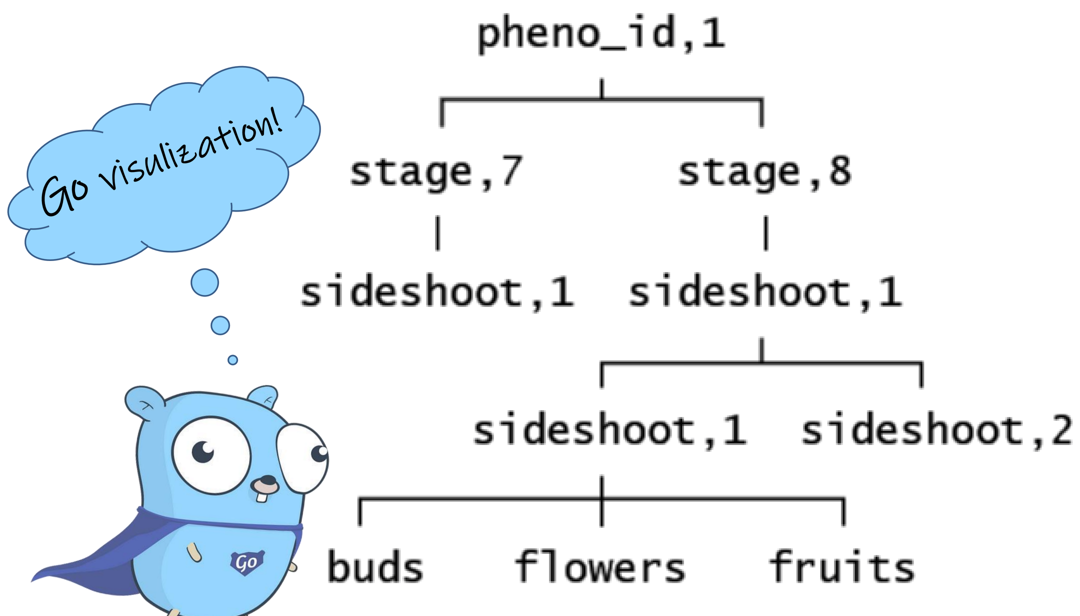
Phenotyper V1.0

plant_id,1	pheno_id,1	pheno_id,2	pheno_id,3
	height internodes leaf_sum pheno_date ...	height internodes leaf_sum pheno_date ...	height internodes leaf_sum pheno_date ...

Phenotyping index for plant id: 1

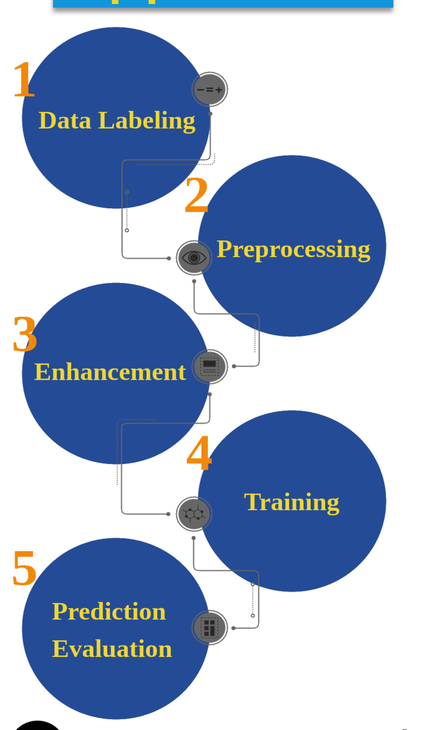
fine_grained_id	pheno_id	location_stages
4	1	8.1
5	1	8.1.1
18	1	7.1
22	2	5.1

Hierarchical structure for side shoots` problem



Example of querying for plant 1 at stage 7 and 8 `*.1`

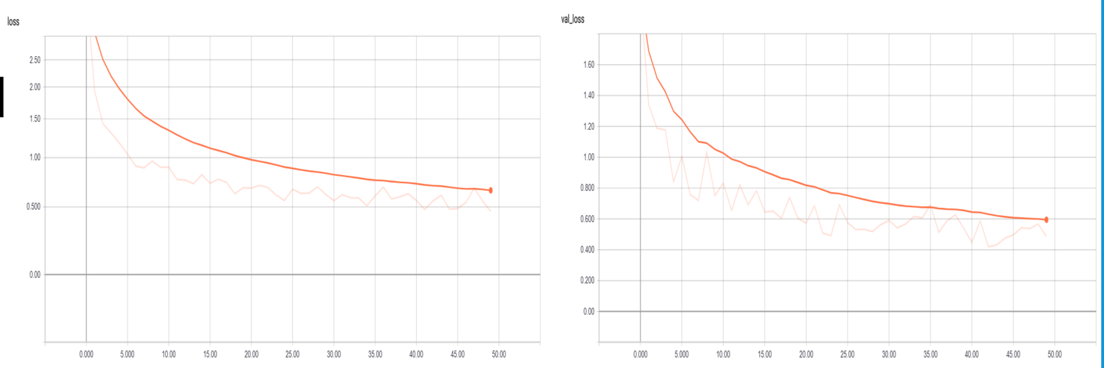
Application




4 angles for image index 284 with the 24th plant and the 8th pose

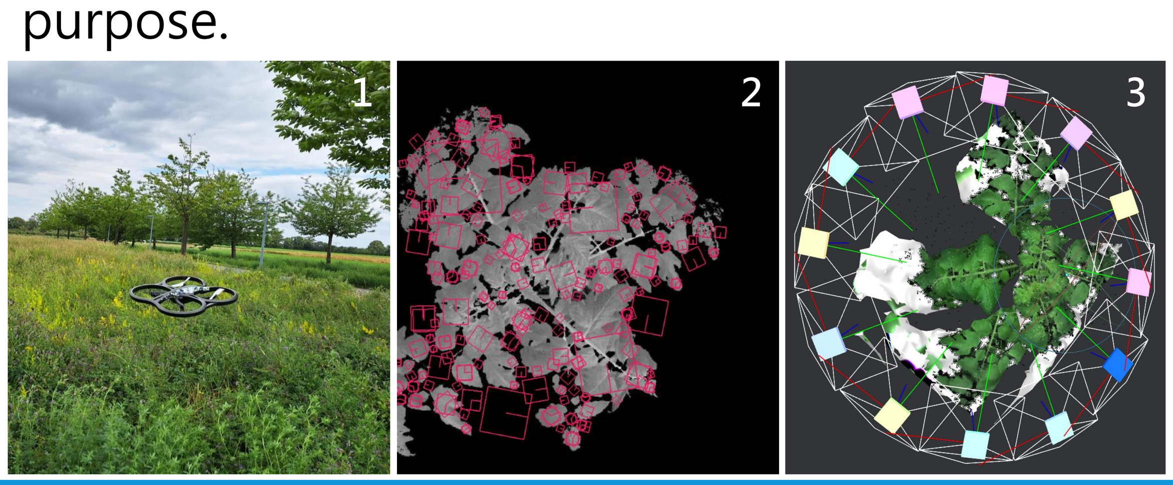

Plant detection and categorization

We annotated and transferred the data into **COCO format** (Common Objects in Context data), and implemented the semantic segmentation and classification approach using **Mask R-CNN** and **VGG-16 backbones.**


Loss vs 50 Epochs

Conclusion

The poster outlines the dataset along with an exceptional approach for fine-grained phenotypic recognition of tomato plants using **deep learning** methods for **precision agriculture** (PA), offering a viable alternative for automated and efficient agricultural practices. Additionally, the innovative **Internet of Things** architecture provides **efficient and scalable data** collecting for phenotyping research purpose.



Node.js programmed drone for **validation data** generation.[1]

3D reconstruction with Phenotyper v1.0 dataset.[2,3]

Acknowledgements:
Sabine Struckmeyer

https://github.com/0YJ/MPTSTD