UNIVERSITÄT BONN

Inaugural-Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Data-driven Image Generation for Crop Growth Modeling

von

Lukas Drees

aus

Bonn, Deutschland

Bonn 2025

Referentin:
Prof. Dr. Ribana Roscher, University of Bonn, Germany

1. Korreferent:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

2. Korreferent:
Prof. Dr. Ian Stavness, University of Saskatchewan, Canada

Tag der mündlichen Prüfung: 24. September 2024

Angefertigt mit Genehmigung der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät der Universität Bonn.

Zusammenfassung

Pflanzenwachstumsmodelle spielen eine entscheidende Rolle in der Landwirtschaft, weil sie dabei helfen zu verstehen, wie Nutzpflanzen wachsen und gedeihen. Sie helfen Landwirten vorherzusagen, wie viel Ernte sie erwarten können, was bei der Planung, Lagerung und Vermarktung von Lebensmitteln wichtig ist. Sie können den Einsatz von Wasser, Dünger und Pflanzenschutzmitteln optimieren, was Kosten spart und zur Nachhaltigkeit beiträgt, indem Umweltauswirkungen minimiert werden. Außerdem spielen sie eine zentrale Rolle in der Forschung, beispielsweise wenn es darum geht, moderne Anbausysteme wie Mischkulturen zu untersuchen und herauszufinden, in welcher Konstellation verschiedene Nutzpflanzen gut gedeihen. Alle Punkte tragen dazu bei, die Landwirtschaft gleichzeitig nachhaltiger und effizienter zu machen, was im Angesicht von Klimawandel und steigender Weltbevölkerung wichtige und zugleich große Herausforderungen sind.

Es gibt viele unterschiedliche Pflanzenwachstumsmodelle, die sich grundsätzlich in prozessbasierte Modelle, datengetriebene Modelle und Mischformen aus beidem eingliedern lassen. In dieser Arbeit liegt der Fokus auf datengetriebenen Modellen, bei denen es darum geht, das Wachstumsverhalten von Pflanzen auf Basis von realen Daten zu lernen. Dazu werden Algorithmen und Verfahren des maschinellen Lernens verwendet. Insbesondere stellen wir zwei grundsätzliche Anforderungen an die Pflanzenwachstumsmodelle, die in dieser Arbeit entwickelt werden. Erstens soll die Modellvorhersage neben anderen Faktoren auf Basis eines Bildes geschehen, das den Status quo der Pflanze in einem frühen Wachstumsstadium zeigt. Zweitens sollen nicht direkt Zielparameter bestimmt werden, sondern zunächst ein künstliches Bild generiert werden, welches ein potenzielles zukünftiges Wachstumsstadium dieser Pflanze zeigt. Aus der Umsetzung beider Anforderungen ergeben sich mehrere Vorteile für die Pflanzenwachstumsmodelle. Die Modellvorhersage beruht so auf realen Daten/Beobachtungen im Feld, wodurch realistische Bilder zukünftiger Pflanzen erzeugt werden können, die einen Zusammenhang zum Input aufweisen. Die generierten Bilder können nicht nur als künstliche Sensordaten weiterverwendet werden, sondern liefern einen erheblichen Mehrwert bei der Visualisierung von

räumlichen Pflanzenverteilungen im Feld und der der Modellerklärbarkeit.

Zur Bildgenerierung nutzen wir die Methode von erzeugenden gegnerischen Netzwerken (Generative Adversarial Networks (GANs)) und führen Experimente auf den Pflanzen *Arabidopsis thaliana*, *Brassica oleracea var. botrytis* (Blumenkohl) und Mischkulturen durch, die aus *Triticum aestivum* (Sommerweizen) and *Vicia faba* (Ackerbohne) bestehen. Wir demonstrieren zunächst, dass ein datengetriebenes Wachstumsmodell, welches ausschließlich auf RGB-Bildern und einem fest definierten Wachstumsschritt basiert, dazu in der Lage ist, realistische Bilder für unterschiedliche Bewirtschaftungsmethoden zu erzeugen. Dabei zeigen wir, dass die Bilder nicht nur realistisch aussehen, sondern sich als künstliche Sensordaten eignen, aus denen sinnvolle Pflanzenmerkmale abgeleitet werden können. Danach erhöhen wir die Input-Flexibilität des datengetriebenen Modells, sodass irreguläre Zeitreihen im Input verarbeitet und Bilder beliebiger Wachstumsstadien erzeugt werden können. Dies ermöglicht nicht nur Inter- sowie Extrapolation von Bildsequenzen, sondern auch die Generierung von stochastischen Bildverteilungen und die pixelweise Visualisierung der Wachstumsvariabilität. Schließlich stellen wir ein drittes datengetriebenes Wachstumsmodell vor, welches einen multi-modalen Input verarbeiten kann, d.h. ein Bild sowie zusätzlich weitere Wachstumseinflussfaktoren verschiedener Art. Es wird gezeigt, dass Ergebnisse eines prozess-basierten Modells als Input für ein datengetriebene Wachstumsmodell genutzt werden zu kann, was genutzt werden kann, um das prozess-basierte Modell eine räumliche Komponente hinzuzufügen oder es zu rekalibrieren. Hervorzuheben ist, dass es gelingt Bilder und Wachstumseinflussfaktoren neu zu kombinieren, wodurch Simulationen möglich sind, die qualitativ und quantitativ analysiert werden.

Insgesamt stellt diese Arbeit signifikante Beiträge zur datengetriebenen Bildgenerierung zum Zwecke der Wachstumsmodellierung dar, indem realistische Bilder von zukünftigen Wachstumsstadien generiert werden, die mitunter mehrere Wochen in der Zukunft liegen und aus denen sich realistische Zielparameter ableiten lassen. Insbesondere die Flexibilität durch irreguläre Bildsequenzen oder multi-modale Bedingungen im Input, die Fähigkeit einen zeitvariablen und stochastischen Output zu generieren und die Integration mit einem Prozess-basierten Modell wird experimentell nachgewiesen. Gepaart mit Untersuchungen über Anforderungen an die Daten sowie Experimenten und Diskussionen zur Generalisierbarkeit der Wachstumsmodelle liefert die Arbeit essentielle Indikatoren, wie bildgenerierende datengetriebene Wachstumsmodelle zukünftig in der landwirtschaftlichen Praxis eingesetzt werden können.

Abstract

Crop growth models play a crucial role in agriculture as they help to understand how crops grow and thrive. They allow farmers to predict how much harvest they can expect, which is important when planning, storing, and marketing food. They can optimize the use of water, fertilizers, and pesticides, which saves costs and contributes to sustainability by minimizing environmental impact. Moreover, they play a central role in research, for example, when investigating modern cultivation systems such as mixed crops and determining in which constellation different crops thrive. These aspects contribute to making agriculture more sustainable and efficient, which are both important and major challenges given climate change and a growing world population.

There are many different crop growth models, which can be categorized into process-based models, data-driven models, and a mixture of both. This thesis focuses on data-driven models, which involve learning the growth behavior of plants from real data. Machine learning algorithms and methods are used for this purpose. In particular, we have two fundamental requirements for the crop growth models that are developed in this work. First, the model prediction should be based on an image showing the status quo of the plant at an early growth stage, in addition to other factors. Second, target parameters should not be determined directly, but an artificial image should be generated first that shows this plant's potential future growth stage. Implementing both requirements results in several advantages for the plant growth models. The model prediction is based on real data/observations in the field, allowing realistic images of future plants to be generated that are related to the input. The generated images can not only be reused as artificial sensor data but also provide significant added value in the visualization of the spatial crop distribution in the field and in model explainability.

For image generation we use the method of Generative Adversarial Networks (GANs) and perform experiments on the plants *Arabidopsis thaliana*, *Brassica oleracea var. botrytis* (cauliflower) and mixed crops consisting of *Triticum aestivum* (spring wheat) and *Vicia faba* (field bean). We first demon-

strate that a data-driven crop growth model based purely on RGB images and a fixed growth step can generate realistic images for different field treatments. We show that the images not only look realistic but can also be used as artificial sensor data from which meaningful plant traits can be derived. Next, we increase the input flexibility of the data-driven model so that irregular sequences can be processed in the input and images of arbitrary growth stages can be generated. This enables not only interpolation and extrapolation of image sequences but also the generation of stochastic image distributions and pixel-wise visualization of growth variability. Finally, we present a third data-driven crop growth model that can handle a multi-modal input, i.e., an image plus additional growth influencing factors of different types. It is demonstrated that results from a process-based model can be utilized as input for a data-driven growth model, which provides the opportunity to add a spatial component to the process-based model or to re-calibrate it. Particularly noteworthy is the ability to recombine images and growth influencing factors, allowing for simulations that are analyzed qualitatively and quantitatively.

Overall, this work contributes significantly to data-driven crop growth modeling by generating realistic images of future growth stages from which realistic target parameters can be derived. In particular, the flexibility through irregular image sequences or multi-modal conditions in the input, the ability to generate a time-variable and stochastic output, and the integration with a process-based model are experimentally demonstrated. In combination with investigations on data requirements and experiments on the generalizability of crop growth models, the work provides essential indicators of how image-generating data-driven crop growth models can be used in agricultural practice in the future.

Acknowledgements

My first gratitude goes to my supervisor, Prof. Ribana Roscher, who has supported me enormously over the last few years. Countless personal meetings with discussions, permanent availability, patience, and understanding in case of failures, extensive feedback and corrections on presentations and papers paired with the freedom to choose my own research focus, and always having an open ear for private matters all speak for an absolutely extraordinary supervision engagement. It speaks for itself that I also have to thank her for my upcoming job at the University of Zurich.

Looking back, my time as a doctoral student would have been unimaginable without the best colleagues in the world, and I would especially like to thank Immanuel Weber for his huge support and for showing me what is important as a doctoral student. Huge acknowledgment to my "office buddies" Jana Kierdorf, Timo Stomberg, Johannes Leonhardt, Eike Bolmer, Ahmed Emam, and Mohammed Farag for really filling the office with life (excluding plants), for great conference experiences together, bringing in new ideas and perspectives, soccer discussions and the countless bike rides. I would also like to thank Julie Krämer, Dereje Demie, and Madhuri Paul for being a fantastic CP5 team. A big thank you goes to Axel Forsch, Armin Billo Corbin, and Sven Gedicke, without whose co-working, including Pomodoro tactics and (many) Geotastic rounds, bike tours, demonstrations, canteen trips and a beer or two, it is questionable whether my work would have been finished at all or a few weeks earlier.

I would also like to thank Prof. Cyrill Stachniss and his group for their warm integration during the first months. In particular, I would like to thank Jens Behley for the outstanding organization of the Reading Group, Thomas Läbe for his extremely patient help with administrative questions, and Birgit Klein for her countless organizational support with simply everything. Many thanks also to Sonja de Vries, Franziska Külbel, Katharina Monaco, Nora Berning, and Maren Kraus for exactly the same, not self-evident support after the move to Niebuhrstraße 1a.

My best friend David Ostendorf deserves special thanks since I can talk to him about anything over an evening of board games. Speaking about board games, the evenings with the wonderful Querbeat-round, especially during Covid, were essential not to lose my head; cheers for that! Great acknowledgement to Anna Schumacher for sharing the last ten wonderful years with me and for her understanding when a "quick" experiment seemed more important than cooking together in the evening. And finally, I would like to thank my parents, Ulrike and Bruno, and sisters, Katharina and Barbara, for always being there for me, without exception.

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2070 – 390732324.

Contents

Chapter 1

Introduction

1.1 Motivation

Due to the fundamental challenges of our time, including the increase in world population, global warming, and the associated consequences such as extreme weather, decline in biodiversity, and the loss of fertile arable land, agriculture is transforming [1]–[3]. To ensure food security in the future, it must become more productive and efficient at the same time, which means using fewer resources such as soil, water, fertilizer, and pesticides while increasing yields [4]. Many approaches that aim to achieve this are coming from the field of smart farming, which uses digital technologies to drive automation through robotics, remote sensing, and machine learning. Applications in this diverse area encompass targeted harvesting, early detection of plant diseases, and low-resource weed removal [5], [6].

The basis for smart farming is generally high-throughput plant phenotyping [7], [8], where structural plant traits, e.g., overall size, canopy cover, degree of maturity, or number and position of fruits and leaves, are recorded and analyzed. Image-based sensors are important for plant phenotyping because they can provide fast, cost-effective, and non-destructive data. Automated imaging systems such as Unmanned Aerial Vehicle (UAV) can capture traits from thousands of plants remotely, significantly speeding up the phenotyping process compared to manual in-field assessments while providing valuable information on plant morphology. Such information can then be used to assess the status quo of fields and initiate targeted in-field interventions that maximize yield under low resource input.

For the modeling of crop growth, which is the core topic of this work, plant phenotyping must be conducted several times during a growth period. It involves comprehensive monitoring and analysis of plant growth determined by

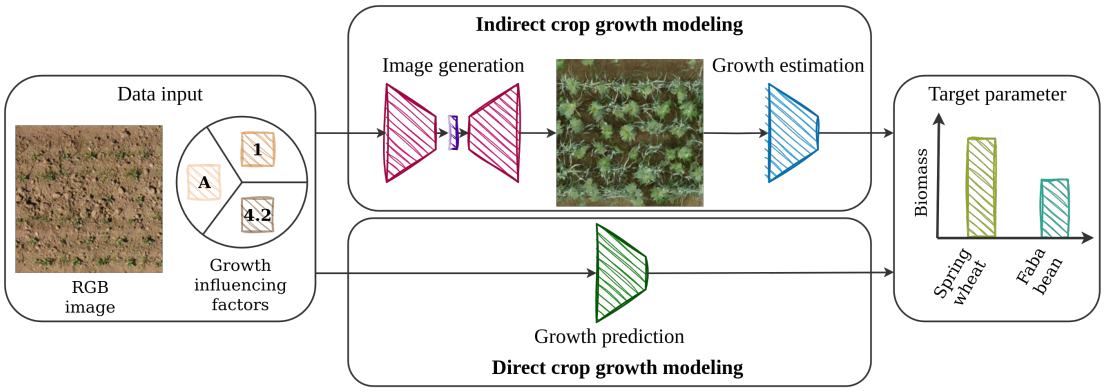


Figure 1.1: Direct and indirect data-driven crop growth modeling in comparison. In the proposed indirect crop growth modeling from the input image and growth influencing factors, first, an image of the later growth stage is generated, and target parameters are derived from this; in direct crop growth, target parameters are predicted directly from the input.

specific growth-relevant parameters, from sowing to harvest [9]. A major motivation is growth prediction, i.e., a temporal prediction and subsequent analysis of future plant traits based on previous observations, prior assumptions, or expert knowledge [10], [11]. Farmers benefit from growth prediction in multiple ways, especially in action adaptation, reliability, and planning. Growth prediction allows early action to positively change the prediction's outcome, for example, through in-field interventions. If the amount and timing of the harvest can be estimated in advance, better conditions can be negotiated with customers, such as supermarkets, at an early stage. In addition, it is very practical in determining when and how much personnel must be hired and how many agricultural vehicles should be rented.

1.2 Main goals

Apart from directly predicting specific plant traits, this work proposes a two-step approach for crop growth modeling as depicted in Fig. 1.1: The first step aims to generate realistic images of probable future above-ground plant phenotypes. For this purpose, we mainly utilize the deep learning concept of GANs[12], suitable for creating new, realistic, and sharp images from given plant image distributions. This step is the focus of this work, including a comprehensive analysis of different generation techniques with different kinds and numbers of growth influencing factors considered. In the second step, we derive agronomically relevant traits, such as projected leaf area or plant biomass, from these generated images through plant phenotyping. This indirect two-step approach has three key advantages over direct plant growth modeling:

- Artificial sensor data. The generated image can be treated as or even re-

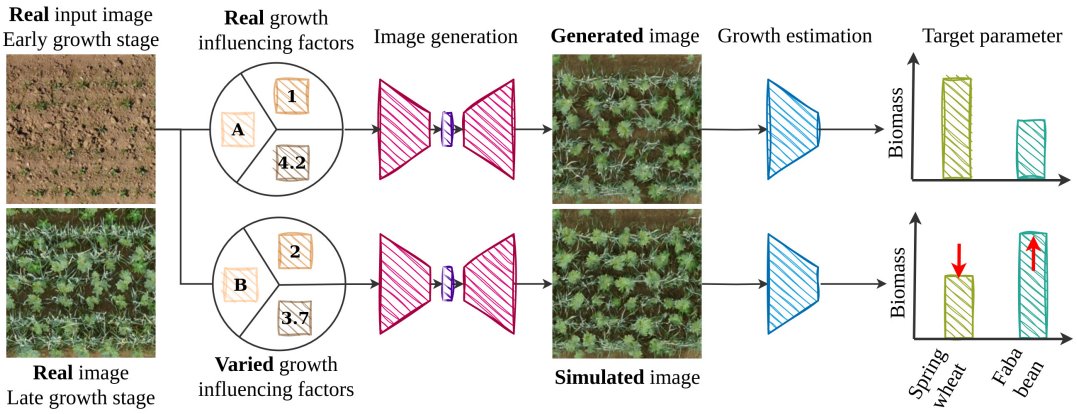


Figure 1.2: Distinction between image generation (top row), in which images are generated with the actual associated multi-modal (categorical, discrete, and continuous) growth influencing factors, and simulation (bottom row), in which these factors are varied to analyze the effects on generated images and target parameters.

place real sensor measurement. Without the need to develop multiple Crop Growth Models (CGMs) for different plant traits, various target traits can be flexibly derived from the generated image of the target growth stage. This simplifies plant phenotyping, as we can directly work with the growth stage we are interested in.

- Spatial plant distribution. Images, especially taken from a birds-eye view, provide a valuable overview of the 2D spread of plants over the ground. As crop fields are often heterogeneous, weeds, pests, and nutrient deficiencies appear more likely in some field regions than in others. Images can identify these regions, and even the development of specific affected plants or leaves can be visualized, enabling targeted in-field interventions in the first place.

- Explainability. Visualizations of target traits, as images do, build reliability in the CGM. Plant growth is a very complex process, and it is not always clear how certain factors influence growth and why the model output is as it is, particularly when only specific target traits are predicted. With direct plant growth modeling, it is not necessarily understandable why a model output changes a certain way when the input changes. Generated images as an intermediate step can thus help to increase the model explainability [13].

Both steps, image generation and phenotyping, will be applied to different time series data sets with different levels of complexity. From the controlled laboratory environment, where the plant *Arabidopsis thaliana* is analyzed, to real field conditions investigating *Brassica oleracea var. botrytis* (cauliflower) and crop mixtures, which consist of *Triticum aestivum* (spring wheat) and *Vicia faba* (faba

Table 1.1: Overview of which input conditions are used in which image generation model. [1]Time is integrated into the modeling, but implicitly by arranging the data into image pairs of different growth stages, not explicitly by model design.

Name	Time	Single image	Multiple images	Infl. factors	Chapter
Paired image to image translation	~[1]	✓	×	×	Chap. 5
Inter- and extrapolating of irregular image time series	✓	×	✓	×	Chap. 6
Multi-modal image generation and simulation	✓	✓	×	✓	Chap. 7

bean) sown under varying treatments. We aim to show that the data-driven CGM becomes increasingly flexible, realistic, and useful across different plants as more conditions, depicted in Tab. 1.1, are incorporated. Starting from an existing plant image at a different growth stage, we add the factor time along with an irregular image sequence to obtain higher prediction flexibility and handle irregular time series before including multi-modal growth-influencing variables, such as the plant cultivar or the seed density.

For simulation purposes, we aim to change the constellation of growth influencing factors for inference, as shown in Fig. 1.2 to analyze the change in the corresponding generated images and derived target parameters. By additionally integrating a stochastic part into the generation process, a diverse set of images for each point in time can be generated - even with a constant input image and constant growth influencing factors.

Without image output, simulations are also possible with well-established process-based CGMs. This thesis demonstrates how the output of such a process-based model can be linked to a data-driven image generation model. Since more conditions mean higher data requirements and thus costs, there is a trade-off between requirements and the accuracy of data-driven plant growth modeling.

1.3 Main contributions

The process of data-driven image generation poses several challenges, partly originating from the specific complexities in the agricultural sector. The following describes the main challenges and what is contributed to addressing them.

Sequential non-equidistant input and time-series output. It is often beneficial to predict growth not only for a single time point but flexibly for any

number of target time points, i.e., different images of different growth stages are to be generated with the same model: an image time series. The requirement for such a time series is, first, that a realistic growth development is recognizable, second, that temporally consecutive images are consistent. Similar flexibility should also apply to the input of image generation: If several images of different plant growth stages already exist, it should be possible to use them jointly to give the CGM multiple image support points. However, due to irregular observations, the input is usually a short, non-equidistant image sequence with different time intervals rather than a complete image time series. Explicit integration of the factor time in the image generation model is necessary for handling sequences in the input and specifically generating time points in the output, resulting in an image time series. We propose using sinusoidal positional encoding and show that a realistic image time series can be generated after processing sequential input with a combined CNN-transformer encoder module.

Multi-modal conditions. Plant growth modeling is highly complex, underlying a multitude of growth-influencing variables. In standard methods of conditional image generation, however, this complexity does not occur in the conditions; it is controlled only by individual conditions, e.g., either by class labels or images, but not both together. The integration of multi-modal conditions is especially difficult because the conditions come with different data types and, therefore, have to be processed differently: class labels as categorical variables (e.g., different crop varieties), time points as discrete variables, measured values as continuous variables (e.g., biomass values), and input images as 3-dimensional tensors. We present several methods to integrate and link these conditions for crop growth modeling, from individual embeddings and subsequent concatenation in latent space to more sophisticated methods such as conditional normalization. Starting with the most important condition, time, we show that a temporal gap between input and output can be bridged generally before gradually adding more growth-influencing variables. In addition to these variables, we also present a way to link results from dynamic process-based models with a data-driven growth model.

Visible variability. The whole process of plant growth is subject to some unknown uncertainty. Even if all growth-influencing factors are identical, plants will differ due to random variations. This is reflected in the data uncertainty, which, together with the model uncertainty, contributes to the overall predictive uncertainty. While this thesis does not aim to separate data and model uncertainty, the resulting visual variability between generated samples should still be realistic. A major problem in image-conditioned image generation is that the stochastic part is often suppressed during generation, leading to a purely deterministic, non-diverse, and overconfident output up to mode collapse. We alleviate

the problem by exploiting optimization with the Wasserstein distance in a multi-conditional context, partially complemented with further loss terms added to the total objective function.

Plant trait based evaluation. The issue of how to comprehensively evaluate generated images is unresolved. There are various evaluation metrics, each with its own advantages and disadvantages, and a combination of metrics must always be used to assess the quality of the generated images. The special challenge of image generation in the context of plant growth modeling is that the generated images should appear realistic and appealing and be suitable for phenotyping tasks in a second step, i.e., serve as artificial sensor data. For this purpose, we use an individual plant trait-based evaluation for each dataset, such as estimating projected leaf area or biomass. This often provides more meaningful and relevant information about the generated image than classical evaluation metrics.

Altogether, the work includes four contributions that improve data-driven image generation for plant growth modeling, namely (1) the handling of sequential non-equidistant input and the generation of consistent output time series, (2) the integration of multi-modal conditions, (3) the generation of a realistic output distribution and thereby variability for each growth stage, and (4) the evaluation of generated images with data-specific plant-traits. Two frameworks are publicly available at

Two frameworks are published open source, containing data-driven CGMs based on sequential input (Chap. 6) and based on multi-modal conditions (Chap. 7). In addition, two crop mixture RGB image datasets collected within the PhenoRob project in 2020 showing pre-processed field patches are published open-source on the PhenoRoam platform.

- **TransGrow**, Python, presented in Chap. 6
  `https://github.com/luked12/transgrow`

- **CGANs for Crop Growth Simulations**, Python, presented in Chap. 7
  `https://github.com/luked12/crop-growth-cgan`

- **Mixed-CKA**, Sequential RGB image dataset, introduced in Chap. 4
  `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/751c10c4-b6dc-4bcc-bc8c-c0fc5920887a`

- **Mixed-WG**, Sequential RGB image dataset, introduced in Chap. 4
  `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/d9d0434f-7864-435e-9c75-56102d9332cb`

1.4 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles, for which I have been the main contributor:

- L. Drees, L. V. Junker-Frohn, J. Kierdorf, and R. Roscher, "Temporal prediction and evaluation of brassica growth in the field using conditional generative adversarial networks," *Computers and Electronics in Agriculture*, vol. 190, p. 106 415, 2021, ISSN: 0168-1699. DOI: 10.1016/j.compag.2021.106415

- L. Drees, I. Weber, M. Rußwurm, and R. Roscher, "Time dependent image generation of plants from incomplete sequences with cnn-transformer," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, and I. Ihrke, Eds., Cham: Springer International Publishing, 2022, pp. 495–510, ISBN: 978-3-031-16788-1. DOI: 10.1007/978-3-031-16788-1_30

- L. Drees, D. T. Demie, M. R. Paul, J. Leonhardt, S. J. Seidel, T. F. Döring, and R. Roscher, "Data-driven crop growth simulation on time-varying generated images using multi-conditional generative adversarial networks," *Plant Methods*, vol. 20, no. 1, p. 93, Jun. 2024, ISSN: 1746-4811. DOI: 10.1186/s13007-024-01205-3

Different approaches to this work were part of different collaborations, which we have acknowledged in the individual chapters and led to the following peer-reviewed publications:

- J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, and R. Roscher, "Behind the leaves: Estimation of occluded grapevine berries with conditional generative adversarial networks," *Frontiers in Artificial Intelligence*, vol. 5, 2022. DOI: 10.3389/frai.2022.830026

- J. Leonhardt, L. Drees, P. Jung, and R. Roscher, "Probabilistic biomass estimation with conditional generative adversarial networks," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, and I. Ihrke, Eds., Cham: Springer International Publishing, 2022, pp. 479–494, ISBN: 978-3-031-16788-1. DOI: https://doi.org/10.1007/978-3-031-16788-1_29

- Q. Marashdeh, L. Drees, and R. Roscher, "Semantic uav image segmentation of mixed cropping fields," in *Proc. of the Dreiländertagung der DGPF, der OVG und der SGPF in Dreseden - Publikationen der DGPF*, vol. 30, 2022, pp. 140–148

- M. Miranda, L. Drees, and R. Roscher, "Controlled multi-modal image generation for plant growth modeling," in *Proc. of the IEEE International Conference on Pattern Recognition (ICPR)*, 2022, pp. 5118–5124. DOI: 10.1109/ICPR56361.2022.9956115

- J. Leonhardt, L. Drees, J. Gall, and R. Roscher, "Leveraging bioclimatic context for supervised and self-supervised land cover classification," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, 2023. DOI: 10.1007/978-3-031-54605-1_15

There are also publications in which I was involved which are not part of the thesis:

- R. Roscher, L. Drees, and S. Wenzel, "Sparse representation-based archetypal graphs for spectral clustering," in *Proc. of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2017, pp. 2203–2206. DOI: 10.1109/IGARSS.2017.8127425

- L. Drees, R. Roscher, and S. Wenzel, "Archetypal analysis for sparse representation-based hyperspectral sub-pixel quantification," *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 5, pp. 279–286, 2018. DOI: 10.14358/PERS.84.5.279

- L. Drees, J. Kusche, and R. Roscher, "Multi-modal deep learning with sentinel-3 observations for the detection of oceanic internal waves," in *Proc. of the ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-2-2020, 2020, pp. 813–820. DOI: 10.5194/isprs-annals-V-2-2020-813-2020

- R. Roscher, M. Volpi, C. Mallet, L. Drees, and J. D. Wegner, "Semcity toulouse: A benchmark for building instance segmentation in satellite images," *Proc. of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, pp. 109–116, 2020. DOI: 10.5194/isprs-annals-V-5-2020-109-2020

- J. Kierdorf, T. T. Stomberg, L. Drees, U. Rascher, and R. Roscher, "Investigating the contribution of image time series observations to cauliflower harvest-readiness prediction," *Frontiers in Artificial Intelligence*, vol. 7, Sep. 2024, ISSN: 2624-8212. DOI: 10.3389/frai.2024.1416323

Chapter 2

Basic techniques

In this chapter, the basic techniques of this thesis are presented, including the taxonomy of generative modeling, the notation used in this thesis, methods on which subsequent models are based or serve as a baseline, and metrics to evaluate generated images.

2.1 Taxonomy of generative modeling

Statistical models can be primarily divided into two classes, discriminative and generative models. While discriminative models draw decision boundaries in the data space and are thus suitable for classification and regression tasks, generative models aim to determine or approximate the data distribution [27]. Hence, generative models are suitable for clustering, representation learning, and density estimation of a data distribution. In addition, new data instances can be sampled from the modeled probability density, which is the key property of why this thesis focuses on generative models.

Given discrete data x, the modeling of the probability distribution $P_{\text{data}}(x)$ can be completely unsupervised. Likewise, with the addition of labels y, the conditional probability distribution $P_{\text{data}}(x \mid y)$ can be captured. The challenge of determining the best models to represent or approximate the data distribution with the global optimum $P_{\text{data}} = P_{\text{model}}$ is subject of current research and highly dependent on the data type. For image distributions, model parameterization is not an easy and intuitive task since images with spatial, channel, and (often) temporal dimensions generally represent high-dimensional data. For this reason, deep neural networks with thousands of parameters are often used and predestined through end-to-end training to approximate image distributions. The use of deep neural networks for modeling these distributions is called deep generative modeling in this thesis.

Considering the type of density estimation, generative models can be generally divided into two types: While explicit density estimation tries to define $P_{\text{model}}(x)$ explicitly, implicit density estimation tries to generate samples that come from $P_{\text{model}}(x)$ but without defining it explicitly beforehand. Examples of explicit density estimation include autoregressive models [28], energy-based models [29], Variational Autoencoders (VAEs) [30], flow-based generative models [31], and denoising diffusion probabilistic models [32], while GANs [12] are examples of implicit density estimation. Some of the above models, specifically VAEs and GANs, are used in this work and will be considered in more detail below. Specifically, we consider how these models can be extended to capture the conditional probability, which is highly relevant in crop growth modeling if additional influencing factors (conditions) are given.

2.2 Notation

In this thesis, images are notated with $\boldsymbol{X} \in \mathbb{R}^{W \times H \times C}$, having a width of W, height of H, and channel depth of C. Unless otherwise specified, $C = 3$ is applied in this thesis, which means images have RGB channels by default. Typically, they are combined into a dataset comprising a total of N images. Thereby we consider three different types of datasets:

- Classic image dataset $\mathcal{X}$: There is the classic variant of an image dataset with $\mathcal{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N]$.
- Paired image dataset $\mathcal{P}$: This dataset consists of two subsets $\mathcal{P} = [{}^{\mathrm{A}}\mathcal{P}, {}^{\mathrm{B}}\mathcal{P}]$ containing aligned image pairs of two domains A and B. Both subsets ${}^{\mathrm{A}}\mathcal{P} = [{}^{\mathrm{A}}\boldsymbol{X}_1, {}^{\mathrm{A}}\boldsymbol{X}_2, \ldots, {}^{\mathrm{A}}\boldsymbol{X}_M]$ and ${}^{\mathrm{B}}\mathcal{P} = [{}^{\mathrm{B}}\boldsymbol{X}_1, {}^{\mathrm{B}}\boldsymbol{X}_2, \ldots, {}^{\mathrm{B}}\boldsymbol{X}_M]$ represent images from one domain each with $M = N/2$. So an image pair is given by $[{}^{\mathrm{A}}\boldsymbol{X}_m, {}^{\mathrm{B}}\boldsymbol{X}_m]$. Different domains here are represented by plants of two different growth stages, which means a temporal domain gap.
- Sequential image dataset $\mathcal{S}$: The sequential image dataset contains K image sequences $\mathcal{S} = [{}^{1}\mathcal{S}, {}^{2}\mathcal{S}, \ldots, {}^{\mathrm{K}}\mathcal{S}]$, whereby a sequence contains ${}^{k}J$ aligned images over time ${}^{k}\mathcal{S} = [{}^{k}\boldsymbol{X}_1, {}^{k}\boldsymbol{X}_2, \ldots, {}^{k}\boldsymbol{X}_{kJ}]$ and associated times ${}^{k}\boldsymbol{t} = [{}^{k}t_1, {}^{k}t_2, \ldots, {}^{k}t_{kJ}]$. Since the number of images per sequence can vary, $N = \sum_{k=1}^{K} {}^{k}J$ is the total number of images in the dataset. One sequence here represents images of the same plant at different growth stages.

In general, $\mathcal{P}$ and $\mathcal{S}$ can be drawn from $\mathcal{X}$ or from parts of $\mathcal{X}$ if the requirements of pairwise resp. sequential image alignments are fulfilled. In all three types of datasets, additional information may be present on a per-image, per-image-pair,

or per-sequence basis. This information is generally notated as y but may have different dimensions and is precisely defined in each specific experiment. During modeling, images are usually further divided into input images $\boldsymbol{X}^{\text{in}}$, those images that the model uses as input, generated images $\boldsymbol{X}^{\text{gen}}$, the model output, and reference images $\boldsymbol{X}^{\text{ref}}$, with which generated images are evaluated.

Major model components frequently used in this thesis are the generator $\mathcal{G}_\theta$ and the discriminator $\mathcal{D}_\delta$, which represent neural networks with corresponding parameters θ and δ. Furthermore, an encoder $\mathcal{Q}$ is used to encode data into a compressed latent space $\boldsymbol{z}$ and a decoder $\mathcal{P}$ to decode the latent space back to the data space, where $\mathcal{Q}$ and $\mathcal{P}$ also represent neural networks. A random noise vector notated as $\boldsymbol{\epsilon}$ is often used as stochastic model input and sampled from the standard normal distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$ unless otherwise specified.

2.3 Variational autoencoder

Variational Autoencoders [30], as explicit density estimators, aim to represent the underlying features of the data in a compact, meaningful, and probabilistic space from which new samples can be generated. This space, generally less dimensional than the original data, is called latent space $\boldsymbol{z}$. To build the latent space, VAEs are based on the idea of Autoencoders (AEs) [33], which try to reconstruct the original data. Here, data is first encoded with $\boldsymbol{z} = \mathcal{Q}_\phi(\boldsymbol{X})$ and second decoded back from it with $\boldsymbol{X}^{\text{rec}} = \mathcal{P}_\psi(\boldsymbol{z})$ second. So an overall objective can be formulated as L2 reconstruction loss between input and reconstructed image.

$$\mathcal{L}_{\text{AE}}(\phi, \psi; \boldsymbol{X}) = \|\boldsymbol{X} - \mathcal{P}_\psi(\mathcal{Q}_\phi(\boldsymbol{X})\|^2 \tag{2.1}$$

Since only the reconstruction loss is used, which leads to a deterministic character of the AE latent space, it lacks two essential properties: First, there is no guarantee that two images close to each other in the latent space will look similar after decoding (continuity). Second, there should be no point in latent space that is not meaningfully decoded (completeness). Both come from the fact that we have no modeling control over how the latent space behaves in those areas where no data samples fall. Reconstructions are therefore possible, but an AE can not be considered a generative model.

With VAE, both properties can be achieved by setting the condition of a prior distribution $P(\boldsymbol{z})$ to the latent space and thus turning it probabilistic. In most cases $P(\boldsymbol{z})$ corresponds to the normal distribution $\mathcal{N}(0, 1)$. To achieve this, instead of directly encoding $\boldsymbol{z}$, the encoder output for each sample is a multivariate Gaussian parameterized by $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. From this, $\boldsymbol{z}$ could be sampled, however, sampling is not a differentiable operation and would therefore prevent gradients

from flowing to the encoder during the backpropagation. Instead, $\boldsymbol{z}$ is calculated via a reparametrization trick, where $\odot$ represents the Hadamard product and $\boldsymbol{\epsilon} \sim P(\boldsymbol{z})$.

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \tag{2.2}$$

An additional regularization needs to be applied, which forces each sample to follow $P(\boldsymbol{z})$. For this, the Kullback-Leibler (KL) divergence is used.

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}(\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X}) \| P(\boldsymbol{z})) \tag{2.3}$$

The overall objective function for VAE finally results in

$$\mathcal{L}_{\mathrm{VAE}}(\phi, \psi; \boldsymbol{X}, \boldsymbol{z}) = \mathbb{E}_{\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X})}[\log \mathcal{P}_\psi(\boldsymbol{X}|\boldsymbol{z})] - D_{\mathrm{KL}}(\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X}) \| P(\boldsymbol{z})) \tag{2.4}$$

where the first part corresponds to a reconstruction loss and the second part to the KL divergence loss. In terms of an interpretable latent space $\boldsymbol{z}$, it is preferable if each dimension in $\boldsymbol{z}$ specifically represents a particular feature. This characteristic is called latent space disentanglement and is an important property for controlled image generation. For instance, it allows to control of plant size, orientation, and color independently. A prerequisite for this are uncorrelated dimensions in $\boldsymbol{z}$, which is not automatically the case with classical VAE. To achieve this, β-VAE adds an additional hyperparameter (β) to weight the KL term of the objective function.

$$\mathcal{L}_{\beta-\mathrm{VAE}}(\phi, \psi; \boldsymbol{X}, \boldsymbol{z}, \beta) = \mathbb{E}_{\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X})}[\log \mathcal{P}_\psi(\boldsymbol{X}|\boldsymbol{z})] - \beta D_{\mathrm{KL}}(\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X}) \| P(\boldsymbol{z})) \tag{2.5}$$

Setting the hyperparameter β is a trade-off between reconstruction accuracy and latent space disentanglement. Hence, the smaller β, the better the reconstruction accuracy due to a lower prioritization of the KL term. Conversely, a larger β encourages a larger latent space disentanglement and, thus, a more interpretable latent space. However, there is no means of controlling which feature is encoded at which position in the latent space. This can be discovered by encoding images with concise features and then examining their position in the latent space. Then, enabled by the completeness property, it is possible to interpolate between two points in the latent space and thus generate images with smooth transitions between certain features. We call this image generation by analyzing and interpreting the latent space implicit since there is no way to explicitly generate an image with certain characteristics.

2.3.1 Conditional variational autoencoder

Conditional Variational Autoencoders (CVAEs) enable the explicit generation of certain features if they were included as a condition in the image generation model.

Thereby, the condition y is considered both in the encoder and in the decoder, whereby there are several possible fusion techniques of image and condition.

$$\mathcal{L}_{\text{cVAE}}(\phi, \psi; \boldsymbol{X}, \boldsymbol{z}, y) = \mathbb{E}_{\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X},y)}[\log \mathcal{P}_\psi(\boldsymbol{X}|\boldsymbol{z}, y)] - D_{\text{KL}}(\mathcal{Q}_\phi(\boldsymbol{z}|\boldsymbol{X}, y)\|P(\boldsymbol{z}|y)) \quad (2.6)$$

Remarkably, CVAE create a latent space in which the condition itself is not encoded: since they are already added to the encoder and decoder, a conditional representation in the latent space is obsolete.

2.4 Generative adversarial networks

Another type of deep generative model is a GAN. With GANs, the model density is not to be represented explicitly, as with VAEs by a latent space $\boldsymbol{z}$. Instead, GANs aims to model the data distribution by generating samples of that distribution. However, this is not a trivial task, as the data distribution for images is very complex and inaccessible. Therefore, the solution is to sample from something less complex, like random Gaussian noise, and then learn a generative model that transforms this simpler distribution into realistic instances of the actual data distribution. GANs can therefore be regarded as distribution transformers. To enable this transformation, a GAN consists of two neural networks that act adversarially to each other. First, the generator uses the input random noise sampled from a normal distribution to generate a new data instance. In the literature, these generated instances are also called fake or artificial. Second, a discriminator attempts to classify real and generated instances of the data distributions. This leads to the following objective function

$$\mathcal{L}_{\text{GAN}}(\theta, \delta; \boldsymbol{X}, \boldsymbol{\epsilon}) = \mathbb{E}_{\boldsymbol{X}}[\log(\mathcal{D}_\delta(\boldsymbol{X}))] + \mathbb{E}_{\boldsymbol{\epsilon}}[\log(1 - \mathcal{D}_\delta(\mathcal{G}_\theta(\boldsymbol{\epsilon})))] \quad (2.7)$$

with adversarial optimization for both parts of the GAN

$$\theta^*, \delta^* = \arg\min_\theta \arg\max_\delta \mathcal{L}_{\text{GAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta) \quad (2.8)$$

since it must be minimized with reference to the generator's parameters and maximized with reference to those of the discriminator. This objective function minimizes the Jensen-Shannon divergence between the real and the generated data distribution [12]. The training process involves an iterative alternate update of generator and discriminator weights, in the optimal case, up to the state of Nash equilibrium. The better the generator succeeds in transforming random noise into a realistic sample, the worse the classification of the discriminator becomes. So the Nash equilibrium represents a state where the generator generates samples that are realistic enough to fool the discriminator; it cannot distinguish between real and generated samples.

In fact, GAN training poses great challenges, as it usually involves issues such as vanishing gradients, mode collapse, and training instabilities. Vanishing gradients occur if the discriminator gets too strong and the generator fails to fool the discriminator. As a result, the generator cannot improve because it does not receive sufficient information from the discriminator; the generated images get less realistic. While vanishing gradients are caused by a discriminator that is too strong, mode collapse can be caused by a discriminator that is too weak or stuck at a local minimum. In this case, the generator always tends to generate the same or a limited diverse output, so only a subset of modes of the data distribution. The discriminator also recognizes it as such, but since the generator slightly beats any discriminator optimization, the next epoch will again show a limited set of modes. In addition, since GANs are very sensitive to architecture design, weight initialization, and the choice of hyperparameters, the dynamic adversarial training process is often difficult to stabilize. However, regularization techniques, additional loss functions, or a different optimization, like with Wasserstein Generative Adversarial Networks (WGANs), can help to mitigate the aforementioned issues.

2.4.1 Wasserstein generative adversarial networks

WGANs can be seen as an extension of traditional GANs with an improved optimization technique. The primary difference between traditional GANs and WGANs [34] is the divergence measure used in their objective functions. WGANs use the Wasserstein distance, also known as Earth Mover's Distance, instead of the Jensen-Shannon divergence, used in GANs.

$$\mathcal{L}_{\text{WGAN}}(\theta, \delta; \boldsymbol{X}, \boldsymbol{\epsilon}) = \mathbb{E}_{\boldsymbol{X}}[\mathcal{D}_\delta(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{\epsilon}}[\mathcal{D}_\delta(\mathcal{G}_\theta(\boldsymbol{\epsilon}))] \tag{2.9}$$

$$\theta^*, \delta^* = \arg\min_\theta \arg\max_\delta \mathcal{L}_{\text{WGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta) \tag{2.10}$$

While in classical GANs, the discriminator outputs a probability score for each input, indicating the likelihood of the input being real, in WGANs, the discriminator output is not constrained to a specific range. Since the output can no longer be interpreted as a probability, the discriminator is often called critic in WGANs, but for simplicity, we will keep calling it discriminator. WGANs also need a Lipschitz continuity constraint on the discriminator to guarantee its continuity and thus the existence of the Wasserstein distance [34]. The Lipschitz constraint is enforced through discriminator weight clipping [34] or gradient penalty [35], then it is referred to as WGAN-GP. With WGANs, the training becomes more stable, and it is easier to decide when to stop the training as the discriminator loss directly provides an interpretable distance measure between the distributions, with the drawback that it often takes longer to converge.

2.4.2 Conditional generative adversarial networks

Conditional Generative Adversarial Networks (CGANs) are an extension of classic GANs incorporating additional information to guide the generation process. This allows for more controlled and targeted image generation. The additional or auxiliary information notated as y can represent anything from class labels over continuous information to images and image time series. A special focus in the central parts of this thesis is to investigate how several conditions of different modalities can be integrated in parallel. Both the generator and the discriminator are conditioned on y, which leads to the following modifications in the objective function for WGANs and Conditional Wasserstein Generative Adversarial Networks (CWGANs).

$$\mathcal{L}_{\text{CGAN}}(\theta, \delta; \boldsymbol{X}, \boldsymbol{\epsilon}, y) = \mathbb{E}_{\boldsymbol{X},y}[\log(\mathcal{D}_\delta(y, \boldsymbol{X}))] + \mathbb{E}_{\boldsymbol{\epsilon},y}[\log(1 - \mathcal{D}_\delta(y, \mathcal{G}_\theta(\boldsymbol{\epsilon}, y)))] \quad (2.11)$$

$$\mathcal{L}_{\text{CWGAN}}(\theta, \delta; \boldsymbol{X}, \boldsymbol{\epsilon}, y) = \mathbb{E}_{\boldsymbol{X},y}[\mathcal{D}_\delta(y, \boldsymbol{X})] - \mathbb{E}_{\boldsymbol{\epsilon},y}[\mathcal{D}_\delta(y, \mathcal{G}_\theta(\boldsymbol{\epsilon}, y))] \quad (2.12)$$

The adversarial training scheme remains unchanged so that Eq. 2.8 for CGAN training and Eq. 2.10 for CWGAN training can still be applied. It should be noted that conditional GANs cannot be considered an unsupervised learning approach, as conditions y must be present in the training for each image $\boldsymbol{X}$.

2.5 Evaluation measures for generated images

Human perception of image quality is subjective and can vary considerably between individuals. Therefore, there are a variety of evaluation measures [36] for generated images that aim to approximate human perception as closely as possible while retaining an objective character. They all have different pros and cons as they specialize in different evaluation aspects, such as image sharpness, image diversity, and semantic correctness. This means that several evaluation measures must always be combined to assess the quality of an image generation model comprehensively. However, there is no consensus on which combination of measures is generally best suited - instead, an individually suitable selection must be made for each application. In this work, the measures Mean Absolute Error (MAE), Multi-scale Structural Similarity Index Measure (MS-SSIM), Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [37] are used to directly compare two images, while KL divergence, Wasserstein distance [34], also known as earth mover's distance, and Fréchet Inception Distance (FID) [38] are used to compare image distributions. They are introduced in Sec. 2.5.1. While the direct comparison between two images, generated image vs. real reference image, is not intuitive with unconditional image

generation, as the output cannot be controlled, and therefore, no suitable reference image can be selected, conditional image generation restricts the space of possible outputs. Direct comparisons can thus be performed, especially when generating image pairs or time series. Even if it is not to be expected that generated and reference images match perfectly, a better evaluation measure indicates a better image generation model.

In addition, in the area of plant generation, great importance is attributed to ensuring that images are not only of good quality but that the respective plants are also semantically meaningful, which is not fully covered by the previously mentioned measures. To examine this, realistic dataset-specific plant traits are derived from generated images and compared with plant traits derived from reference images. This plant-trait-based evaluation is generally based on models that are trained independently of the respective image generation model. Especially the basic building blocks of these models, namely Mask R-CNN models for instance segmentation of plants in the images, and a modified ResNet for image-based regression, are introduced in Sec. 2.5.2.

2.5.1 Image quality and distribution measures

The image quality and distribution metrics used are presented in more detail below. Notably, they are not only used for evaluation at test time but can also support model training, e.g., as an addition to GAN's adversarial training loss.

Mean absolute error

The Mean Absolute Error (MAE), also known as the L1 distance, is a well-established image evaluation measure. It is a pixel-wise measure that calculates the absolute differences between corresponding pixels in the generated image and the reference image. The MAE is defined as the average of these absolute differences.

$$\mathrm{MAE} = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} |X_{hw}^{\mathrm{gen}} - X_{hw}^{\mathrm{ref}}| \tag{2.13}$$

Here, H is the height and W is the width of the image. X_{hw} represents the intensity value of the pixel at position $[h, w]$. While MAE is fast to compute, robust to outliers, and well interpretable, it may not fully capture perceptual differences or structural variations between images. Thus, it is often used in conjunction with other evaluation measures.

Peak signal-to-noise ratio

Peak Signal-to-Noise Ratio (PSNR) is a metric used to assess the quality of signal representation, particularly in the context of images and video subject to lossy

compression. Therefore, it is well suited to evaluate the reconstruction quality of generated images. PSNR quantifies the fidelity of a reconstructed signal by comparing it to the original signal and is expressed as a logarithmic quantity using the decibel (dB) scale. In the formula

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{\mathrm{MAX}_\mathrm{I}^2}{\mathrm{MSE}} \right) \tag{2.14}$$

the fraction is calculated between the squared maximal possible pixel value MAX_I, which is 255 for 8-Bit images, and the MSE, also known as the L2 distance and defined as the average of the squared pixel-wise differences.

$$\mathrm{MSE} = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (X_{hw}^{\mathrm{gen}} - X_{hw}^{\mathrm{ref}})^2 \tag{2.15}$$

Typical PSNR values for 8-bit images range between 30 dB and 50 dB, with higher values indicating better reconstruction quality. The advantages of PSNR include its simplicity, which makes it interpretable and accessible to a wide range of users. It's a global measure of image quality, but it is also very sensitive in penalizing image artifacts. So small but high changes in pixel values can lead to a significant drop in PSNR. When aiming for artificial sensor data, this property is particularly important. On the downside, its magnitude depends on the image content, and therefore, PSNR is difficult to compare between datasets. Furthermore, while PSNR is sensitive to pixel-wise differences, it does not capture structural changes in an image. As a result, it performs unevenly for different types of distortions and does not always correlate well with human perception.

Multi-scale structural similarity

For the structural comparison between generated and reference images, we use the Multi-scale Structural Similarity Index Measure (MS-SSIM) [39]. When comparing two identical images, it reaches an optimal value MS-SSIM = 1. MS-SSIM is implemented through a hierarchical approach to evaluate the structural similarity between generated and reference images across different scales aiming to capture both local and global structural information. This involves three steps.

In the first step, both images are decomposed into five different scales by successive 2D average pooling. Structural similarity indices (SSIM) are then computed independently at each scale, comparing window-wise (size 11×11) corresponding sub-regions from the decomposed reference and generated images. This involves assessing luminance, contrast, and structure similarities at both local and global levels, incorporating spatial information within each scale. For each window on each scale, the SSIM is calculated as

$$\text{SSIM} = \frac{2 \cdot \mu_r \mu_g + C_1}{\mu_r^2 + \mu_g^2 + C_1} \cdot \frac{2 \cdot \sigma_{rg} + C_2}{\sigma_r^2 + \sigma_g^2 + C_2}, \tag{2.16}$$

where μ_r μ_g are average intensity values of the windows of the reference or generated image, σ_r^2 and σ_g^2 are the corresponding variances, σ_{rg} is the covariance, and C_1 and C_2 are constants added to the formula to stabilize the division. The window-wise SSIM scores within each scale are averaged and normalized afterward using ReLU activation, which is not compliant with the original definition but stabilizes the metric when used during neural network training. The third step involves aggregating the SSIM scores from different scales, resulting in the final MS-SSIM. The weights are assigned based on the significance of each scale in human perception, since the step aims to emphasize the contributions of scales that are more perceptually relevant, ensuring a balanced and representative measure of overall structural similarity.

Due to the analysis of structural information at different resolutions, MS-SSIM provides a robust and comprehensive method for evaluating image quality, considering the multi-scale nature of human perception. It is particularly useful in capturing perceptual nuances and is often preferred over traditional SSIM for evaluating image quality, especially in scenarios where it's important to capture both fine and coarse details.

Learned perceptual image patch similarity

The Learned Perceptual Image Patch Similarity (LPIPS) [37] metric is a perceptual similarity measure designed to capture the perceived differences between images based on neural network embeddings. For this purpose, the generated and reference images are passed to the same pre-trained network to obtain feature representations ${}^{l}\hat{z}^{\text{gen}}$ and ${}^{l}\hat{z}^{\text{ref}}$ of L layers, which are stacked along the channel-dimension, unit-normalized and scaled along the channel dimension using w_l. Afterward, they are compared using the L2 distance, averaged spatially, and added channel-wise.

$$\text{LPIPS} = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot ({}^{l}\hat{z}^{\text{gen}}_{hw} - {}^{l}\hat{z}^{\text{ref}}_{hw}) \right\|_2^2 \tag{2.17}$$

The lower the LPIPS, the higher the similarity between the two images. Unless otherwise specified, the VGG network [40], a 16-layer deep convolutional neural network developed for large-scale image recognition, is used. Since LPIPS compares high-level features at different latent space scales, it correlates well with human perception. It is also relatively time-efficient, as the weights of the feature extraction network do not need to be trained and is therefore also suitable as a loss function.

Fréchet inception distance

The Fréchet Inception Distance (FID) [38] is an evaluation metric to provide a quantitative similarity between image distributions. Similar to LPIPS, the feature representations of generated and reference images are compared in a pre-trained deep neural network. However, two images are not compared directly over the L2 distance, but statistics are calculated over two sets of images (image distributions). Specifically, the FID is calculated in three steps. First, a pre-trained deep neural network, Inception-v3, is used to extract feature representations from the reference and generated images. The Inception network is usually truncated at the 2048-dimensional pooling layer after the convolutional backbone, one of the deepest layers in the network, and the activations from that layer are used. Second, statistics are calculated for a set of reference and generated images, by defining multivariate Gaussian distributions $\mathcal{N}$ with mean values $\boldsymbol{\mu}$ and covariance matrices Σ of the feature representations. In the third step, the Wasserstein-2 distance between the two Gaussian distributions is computed with

$$\mathrm{FID}(\mathcal{N}_r, \mathcal{N}_g) = \left\|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\right\|_2^2 + \mathrm{tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{\frac{1}{2}}\right), \tag{2.18}$$

where $\mathcal{N}_r(\mu_r, \Sigma_r)$ and $\mathcal{N}_g(\mu_g, \Sigma_g)$ are the Gaussian distributions of reference and generated images respectively. The smaller the $\mathrm{FID}(\mathcal{N}_r, \mathcal{N}_g)$, the higher the similarity between reference and generated distributions.

In contrast to the previously presented metrics, FID is more robust with respect to noise and small perturbations in the images and takes into account (besides the image quality) the diversity of generated images, which makes it sensitive to mode collapse. This means it can detect when the generative model only covers parts of the reference image distribution. Both properties are significant advantages over the Inception Score (IS) [41], which also uses the Inception network but, in contrast to FID, considers the final output layer. Instead of assessing deep features, IS primarily measures the diversity and confidence of the classifier in predicting final class labels, which may be unreliable or biased for plant images because the Inception network was not trained with data from the plant domain. Therefore, FID is more relevant than IS for this thesis.

2.5.2 Plant-trait-based evaluation

Plant-trait-based evaluation is inspired by the idea of deriving plant traits (PT) from images relevant to crop growth analysis. This is first done independently for a generated and a reference image, and subsequently, the derived traits are compared to assess the suitability of the generated image for plant phenotyping purposes. Relevant parameters are, for instance, the size or the extent of the plant, the leaf area, the height, or the biomass. There are different methods for

determining these plant traits, including calculation from vegetation indices, estimation using residual neural networks, and derivation from instance segmentation using Mask R-CNN, which are used in this work.

Plant trait estimation using residual neural networks

The estimation of plant traits (PT) from images falls into the area of image regression. Image regression describes the task of estimating continuous-values outputs from input images. Since deep learning is well suited for image regression in the plant domain, we use residual neural networks (ResNets) [42], one of the convolutional state-of-the-art models, in this work. ResNets, which were originally developed for image classification, are characterized by their residual blocks. Each residual block consists of two main paths: the identity path and the shortcut path. The identity path is the original path through the block consisting of multiple stacks from convolutional layers, batch normalization, and ReLU activations, while the shortcut path directly connects the block's input to its output. This shortcut path enables the network to learn residual functions, allowing it to build and train exceptionally deep networks without vanishing gradients.

There are several ResNet variants from ResNet-18 to ResNet-152, with which the number of layers is indicated. For efficiency reasons - as the number of layers increases, so does the number of parameters and the training time - only ResNet-18 and ResNet-50 are used in this thesis. To make it applicable for regression, it needs to be slightly modified by reducing the final linear layer, which originally had 1000 neurons (the number of classes in Image-Net), to the number of neurons corresponding to the number of target parameters. If, for instance, the biomass in a mixed crop image consisting of two species is to be determined by species, the number of output neurons required is two. The optimization of a ResNet is fully supervised, so pairs consisting of image and regression labels are required. In general, image regression is particularly suitable for tasks where either pixel-wise results are not required or pixel-wise labels are difficult to obtain, e.g., for crops, plant height, biomass, or mixture proportions.

Plant instance assessment using Mask R-CNN

Plant instance assessment is a crucial task in various agricultural applications, where a plant's center point, diameter, or projected leaf area is of interest. Mask R-CNN [43], widely applied in the plant domain, enables instance segmentation, simultaneously performing object detection and semantic segmentation of a plant. Object detection describes the localization of the plant in the image by means of a bounding box and the assignment to a class, whereas semantic segmentation represents the finer pixel-by-pixel segmentation. Both are crucial, as

in purely semantic segmentation, the plant instances cannot be separated if the leaves overlap, which is particularly common in late growth stages. Unlike many related methods, Mask R-CNN streamlines the process by combining bounding box regression, its classification, and semantic mask generation in a parallel head. In addition to state-of-the-art accuracy, the method is thus fast, intuitive, and easy to use. The following are the main steps of a Mask R-CNN: It begins with a backbone Convolutional Neural Network (CNN), such as ResNet or VGG, which processes the input image and extracts a feature map. For object detection, the Mask R-CNN uses a slightly modified Faster R-CNN network [44] for region proposal extraction. In contrast to Faster R-CNN, which uses RoI (Region of Interest) Pooling to extract fixed-size feature maps for each region proposal, Mask R-CNN introduces RoI Align. RoI Align addresses misalignment issues caused by quantization in RoI Pooling, ensuring more accurate pixel-to-pixel alignment between the input image and the extracted features. Each region proposal is then passed through two fully connected layers: one for object classification and another for bounding box regression, where the coordinates of the bounding box are adjusted. This step determines whether the region proposal contains an object and refines its location. For semantic segmentation, a fully convolutional network is used to create binary segmentation masks for each class inside each region proposal. The Mask R-CNN is trained using a multi-task loss function, which combines losses for classification, bounding box regression, and mask prediction. Naturally, the complexity increases the more different classes are used in training, which is why, in this thesis, dataset-specific Mask R-CNNs are developed. There is only one class at a time representing the plant of the dataset, while everything else, including weeds, insects, or field equipment, is considered background.

Quantification of plant traits for a set of images

In the evaluation for a whole test set with N images, the Mean Absolute Error (MAE) and the Mean Error (ME) are calculated as follows between plant traits (PT) of the generated and the reference image.

$$\mathrm{MAE}_{\mathrm{PT}} = \frac{1}{N}\sum_{n=1}^{N}|\mathrm{PT}_n^{\mathrm{gen}} - \mathrm{PT}_n^{\mathrm{ref}}| \tag{2.19}$$

$$\mathrm{ME}_{\mathrm{PT}} = \frac{1}{N}\sum_{n=1}^{N}\mathrm{PT}_n^{\mathrm{gen}} - \mathrm{PT}_n^{\mathrm{ref}} \tag{2.20}$$

Here, the quantity measure ME indicates whether the PT is overall underestimated ($\mathrm{ME} < 0$) or overestimated ($\mathrm{ME} > 0$). For whole agricultural fields, the ME is informative, in case it is not as important to accurately determine the

yield of individual field regions but rather to evaluate whether the overall mean predictive error for the entire field is low.

Chapter 3

Related Work

With the rapid rise of Machine Learning (ML) in general and in particular generative AI in recent years, these technologies have also found their way into the agricultural sector [45]. This chapter examines recent developments and applications related to crop growth modeling and discusses links to this thesis. First, a broader overview of Crop Growth Models (CGMs) is given to understand the agricultural transformation toward data-driven models better. This includes their general importance for agriculture and specifically their relevance for the arable crops cauliflower, wheat, beans, and their mixtures examined in this thesis. In addition, different types of currently used CGMs are presented, along with the advantages and disadvantages of data-driven approaches and their combinations with conventional methods. The following sections cover related work to the data-driven approaches used in this thesis, i.e., recent machine learning and image generation techniques for plant phenotyping and crop growth modeling.

3.1 Importance of crop growth models

Crop growth modeling plays a crucial role in modern agriculture by offering a systematic understanding of the complex interplay of various factors influencing crop development [46]. The ability to predict and simulate crop growth under different environmental conditions is instrumental in optimizing agricultural practices, resource allocation, and crop management strategies. By integrating empirical data and mathematical formulations, CGMs enable researchers and practitioners to make informed decisions regarding irrigation, fertilization, and pest control, ultimately contributing to enhanced crop yields and sustainable agricultural practices [47]. Additionally, these models serve as valuable tools for climate change impact assessment, aiding in developing resilient and adaptive farming systems [48].

3.1.1 Relevance for investigating cauliflower

Modeling cauliflower growth is particularly relevant to accurately predict the harvest-readiness of the curd, i.e., the edible part of the cauliflower head. The general growth development depends primarily on temperature, solar radiation [49], and CO_2 concentration [50]. Since cauliflower is considered a high-value crop in Europe, it is usually not grown under limited water or nutrient supply. Therefore, these parameters are missing in many CGMs [51]. In traditional agricultural practice, the above-mentioned factors are often monitored on parts of a field using spot checks, and the assessment of crop development is then extrapolated to the entire field. However, this method is unsuitable for cauliflower. Here, the assumption that cauliflower heads, planted simultaneously and exposed to the same external conditions, have the same optimum harvest time is not valid. Instead, there is a high within-field variability, equivalent to a long harvest period [51]. At the same time, there is a period of just one week for each plant suitable for harvesting [52]. In addition, the harvesting methods are very labor-intensive, as ripeness can only be determined by touching the curd to check whether it meets size and consistency criteria. If harvested too early, the cauliflower is still underdeveloped, while overripe heads of curds lose their compactness or are exposed to light for too long, resulting in color changes [52]. The combination of the high within-field variability leading to difficult timing and the current need for manual harvesting causes the harvesting operation to account for a large proportion of the total cost of cauliflower production [51]. As cauliflowers are subject to high-quality standards, any deviation from the optimum harvest time will result in a considerable loss of value and disadvantage in the sales market. To summarize, mainly due to the heterogeneity in growth, cost-intensive harvesting, and high-quality demands of the market, the development of CGMs focusing on single heads is of great importance.

3.1.2 Relevance for investigating crop mixtures

Crop mixtures have the potential to increase system productivity compared to sole crops and thus herald more sustainable agriculture in the future [53]. As an example, cereal and legume crop mixtures are known to improve resource use efficiency [54], enhance nutrient acquisition [55], maximize system productivity through complementarity, especially on low input land limited by nitrogen deficiency [56], and reduce weeds, diseases, and insect pest infestations [57]. However, the specific crop responses to complex genotype $\times$ environment $\times$ management interactions are not well explored. In fact, many farmers do not currently consider crop mixtures an option, often due to a knowledge gap in species, cultivar, and treatment selection, which results in performance uncertainty [58].

To gain more knowledge about crop mixture systems, two approaches are suitable: field trials on the one hand and Crop Growth Models (CGMs) on the other. While field trials integrate actual environmental and management conditions, they are limited in time and space and can only test a small number of such conditions. Crop growth modeling, on the other hand, may be limited in predicting realistic responses of crops, especially under a changing climate [59], but allows the simulation of multiple conditions, including future environments. For instance, it can help to simulate when domination occurs, i.e., an unbalanced proportion of biomass in the mixture. This helps to understand the competitive balance between crop mixtures at different growth stages, which is essential for their viability [53].

3.2 Types of crop growth models

The growth of plants is influenced by abiotic factors, including soil composition, temperature, precipitation, and other climate conditions, as well as biotic factors such as pollinators, pests, weeds, and pathogens [60]. Many different conventional models have been developed to predict crop growth based on the aggregation of parts of these factors over the growing period, the estimation of photosynthesis, or different climate conditions [51], [61], [62]. Conventional CGMs can be divided into Process-Based crop growth Models (PBMs) (often also referred to as mechanistic), empirical, and combined variants of both [48]. It is important to distinguish between empirical and Data-Driven crop growth Models (DDMs) used in this thesis. While empirical models are a subset of DDMs, not all DDMs are empirical. In particular, generative ML algorithms, which we categorize as DDMs, learn complex underlying patterns from data and require a high level of optimization, which cannot be considered empirical.

3.2.1 Process-based crop growth models

Process-Based crop growth Models (PBMs) simulate crop growth by representing the dynamic interactions between various components, such as soil, climate, and plant physiology. These dynamic interactions are generally based on physical equations, which describe, for example, water and nitrogen uptake from the soil or the sun-induced fluorescence properties of plant leaves. As plant growth is subject to highly complex processes, some of which are not well understood, PBMs need to make decisions regarding the scale of application and the growth-influencing factors considered. For example, many CGMs can handle the abiotic factors CO_2, water and nitrogen uptake well, while many biotic factors such as weeds and pathogens are only considered in very few models [63]. This leads to

simplifications and biases in the model, meaning that a complex calibration to the current application site is usually required, for which experiments or domain knowledge of realistic parameter bounds are used [63], [64]. In total, PBMs are valuable for understanding the mechanisms influencing crop development and are often used in research and practice for decision support, yield prediction, and optimization of agricultural practices.

Focusing on the farm level, CGMs are usually used to manage irrigation, fertilization, and sowing [63]. Among the most widely used CGMs are AquaCrop [65], APSIM (Agricultural Production Systems sIMulator) [66], both applicable to the optimization of water use efficiency in the context of yield responses of wheat, DSSAT (Decision Support System for Agrotechnology Transfer) [67] simulating the development of crops under different soil, climate and management conditions, and CROPGRO [68] included in the DSSAT system and applicable to model the growth of brassica plants [69]. Specifically for cauliflower, PBMs exist that incorporate relations between the different phases of cauliflower development, crop development, leaf area expansion, increase in curd volume, and increase in dry matter [51]. In this thesis, the SIMPLACE [70] framework is used as PBM to model monocultures and mixtures between wheat and legumes according to various target parameters, including height and biomass.

3.2.2 Data-driven crop growth models

In modern agriculture, the integration of machine learning has revolutionized traditional farming practices by leveraging vast amounts of data to create accurate and dynamic simulations of crop growth. Summarized under the term Data-Driven crop growth Models (DDMs), they include data on crop-specific attributes such as genetics, environmental conditions, and management practices, as well as sensor data. Given the large amount of data fusion techniques in deep learning, there is great flexibility in the use of input data, and no prior information on the relationship between the variables is required [63]. Therefore, they are not dependent on domain knowledge, which can be seen as both an advantage and a disadvantage. On the one hand, they do not benefit from domain knowledge of well-researched processes. The strong data dependency also makes them less generalizable because the field of application is limited to the data on which they were trained [63]. On the other hand, this provides the flexibility to find non-linear patterns and connections about which there may previously have been simplified assumptions or insufficient domain knowledge [7]. In the case of image data, which cannot be easily integrated into PBMs, DDMs have the advantage of accessing additional assumption-free information about the spatial plant development, such as the plant density or the number of plants. This is particularly crucial for mixed crops because the spatial distribution depends on crop emergence and

mixture effects, such as the early possible domination of a partner, for which little expert knowledge is available [53]. Another issue is the low interpretability of ML-models often considered as black boxes since they are trained end-to-end without understanding the parameter relationships in between. This issue is tackled by interpretable machine learning techniques [71] and in this thesis specifically by exploring the latent space of generative models.

There is a variety of empirical/statistical DDMs, suitable for classical regression on weather, growth-influencing variables, and remote sensing data [63]. Recently, ML models are also being used, mostly to predict certain parameters from these data, such as crop yield [72]. Further related work dealing with traits derived from images by ML in this thesis can be found in Sec. 3.3.

3.2.3 Hybrid models

There are also efforts to link both approaches, as they can complement each other, which are hybrid models. Conceptually, for CGMs, the linking can be parallel, serial, or modular, depending on how the data is processed in the two models [73]. In the parallel setting, both models have the same input. In the serial setting, the output of one model represents (parts of) the input of the other model. In the modular case, both models are combined in submodules of the crop growth model. Specifically, several linking strategies are possible: Integrating DDMs into PBMs is possible by using data assimilation methods [74] or ML models to calculate certain parameters of the PBMs in a data-driven way. DDMs can also completely replace certain parts that can be better data-driven modeled than process-based, such as long-term weather forecasts [73]. Conversely, the output or knowledge of PBMs can also be used for DDMs. This follows the idea of theory-guided ML, which, in general, aims to enhance the effectiveness and interpretability of data science by integrating scientific knowledge [75]. For instance, PBMs can generate simulation data that can increase and diversify the training dataset of DDMs. It is also possible to embed knowledge about parameter bounds from PBMs into ML algorithms by adding constraints to the loss function or by choosing certain activation functions in neural networks [73].

Some work already shows that ML techniques can be integrated to PBMs, like the APSIM model, to obtain a higher yield prediction accuracy [76], [77]. However, to our knowledge, no hybrid image-generating crop growth model exists. The focus of this thesis is on combining different data types in DDMs, whereby domain knowledge is included in the integration of this data. For instance, sensor data is integrated differently than growth-influencing variables and the time condition. In addition, a serial interface is created in Chap. 7 to combine the output of a PBM as input of a DDM directly.

3.3 Machine learning for plant phenotyping

Among the DDMs, especially deep learning approaches have had a significant impact on the development of plant phenotyping methods and CGMs in recent years. In particular, CNNs are used for direct target parameter estimation like the assessment or prediction of crop yield and biomass [72]. While the outputs are often one or several one-dimensional target parameters, CNNs are suitable for processing both one-dimensional inputs, such as weather, soil, or treatment data, as well as spatial two-dimensional data, such as plant images. Images have proven very efficient for integrating plant traits into the model, particularly when recorded in a unified orthographic view [78], [79]. By integrating tools for sequential processing, such as recurrent neural networks [80], Long Short-Term Memorys (LSTMs) [81], or transformers [82], the input space is not restricted to a single time point. Hence, there is great flexibility in integrating different data dimensions into neural networks to estimate certain crop growth target parameters.

3.3.1 Estimation of plant traits leaf area and biomass

Accurately determining the actual leaf area from images with an orthographic view is challenging because many leaves are hidden by the top canopy layer. Therefore, either methods such as LiDAR scanning or multi-view imaging, which enable the creation of a 3D model of the plant, or assumptions about the leaf density and height of a plant are necessary for an exact determination.

A widely used alternative to the leaf area is the projected leaf area, which is also used in this work and represents the area of the leaf surface as seen from an orthographic perspective. This can be obtained by segmenting the leaf surface and then counting the segmented pixels with a known Ground Sampling Distance (GSD). There are various options for segmentation, such as the use of vegetation indices [83], semantic image segmentation via CNN [19], or in the case of overlapping instance segmentation via Mask R-CNN [52].

Similar difficulties arise for the biomass, as it depends on the plant volume, which, as a cubic size, cannot be determined exactly from orthographic views without height information. However, adequate estimates of CNNs can actually be made from drone images [78] as well as satellite images [18]. It is expected that the visual characteristics such as the structure, density, leaf size, leaf color, and shadow effects, which all change during the growing season, provide valuable proxies.

3.4 Image generation for plant phenotyping

This thesis proposes first to generate images and then derive important target traits by plant phenotyping afterward. While reference labels of the target traits are required for the second step, the first step of image generation can be regarded as unsupervised learning. This is a major benefit, as the gathering of labeled training data is time-consuming and cost-intensive. In addition, labeled reference data are often related to yield and acquired at the end of the growth period - usually in destructive measurements - while image-based plant phenotyping is non-destructively possible over the entire growth period [84]. Generating plant image data can improve subsequent downstream tasks, but even the generated images themselves bring added agricultural value in many respects.

Plant image datasets often suffer from diversity, as plants exhibit enormous biological variability and often have complex overlapping structures, which are additionally diverse by environmental influences [45], [85]. The generation of artificial sensor data for data augmentation helps to enrich the dataset's diversity by generating new realistic plant images [45], [86]–[88]. Artificially generated images are also relevant for transfer learning by domain adaptation, whereby newly observed plant image datasets are given a style (changed exposure, background) that corresponds to known environments, e.g., from existing pre-trained models[89]. Super-resolution tasks similarly fall into the category of artificial sensor data, where deep learning is used to synthesize a higher-resolution image from a low-resolution one, which is useful for visualizing fine-grained leaf structures and detecting anomalies [90]–[92]. In all cases, the higher diversity and quality of the datasets through augmentation, domain adaptation, and super-resolution help to improve target tasks such as crop and disease detection and classification.

In many scenarios, the generated images can not only be seen as an intermediate step, as they contribute to model understanding and explainability and further create expert knowledge [71]. This can be seen, for instance, in the area of plant diseases, where it is crucial to understand how biotic and abiotic stresses affect plant organs and how diseases progress [93]. Explainability leads to trustworthiness in ML models, which is also enhanced when the probable position of fruits hidden behind leaves is visualized rather than the total number of fruits estimated by regression [17]. Similarly, in crop mixture scenarios, it is useful to visualize spatial development, as this can provide valuable insights into field heterogeneities and different genotype interactions [53]. This enables not only targeted in-field interventions but also leads to a better understanding of how two species compete with respect to certain influencing factors.

3.5 Image generation for crop growth modeling

There are few studies that incorporate the temporal dimension to model the development of plant growth over time using generated images, i.e., create realistic probable estimates of future plant phenotypes.

Hamamoto et al. [94] use classical CGANs for image generation. In the generator, LSTMs are used both in the encoder to handle equally spaced image sequences of earlier growth stages and in the decoder to generate a sequence of future images in one step. They treat crop growth modeling as video representation learning, i.e., the prediction of the next frame from a sequence of previous frames. This is reasonable for greenhouse environments with an enormously high observation rate but is not transferable to real field datasets with usually significantly fewer observations and irregular observation intervals (cf. dataset characteristics in Sec. 4.3). Limitations are short growth prediction steps of hours to a few days and a small image size of 128×128 px, which are both increased in this thesis.

They also present the idea of integrating shape priors in the form of leaf segmentation images into the generation by early fusion with the RGB images in the LSTMs input. This has the advantage of directly obtaining the segmentations of the future plant in the output alongside the predicted RGB images. While these segmentations show an appealing quality in experiments with Komatsuna (*Brassica rapa*), the generated RGB images are very blurry, and the shape priors also do not prove helpful for the generation quality.

Slightly different shape-based prior are used by Kim et al. [95], who show image-based crop growth modeling without GANs using several plant factory datasets, also based on equidistant image sequences as model input. Here, the growth prediction process is divided into first, shape prediction using a spatial transformer network [96] to predict the future plant morphology, and second, RGB reconstruction, in which the new shape image is filled in based on the color information from the last available point in time using hierarchical AE. The prediction steps are short and in the experiments 3 h for Komatsuna and 1 d each for Arabidopsis and Butterhead lettuce (*Lactuca sativa var. capitata*) plants. In general, the shape prior offers the advantage of controlling the growth process and prevents artifacts and unnaturally shaped leaves, as the generated image is an affine transformation of older growth stages. This is at the expense of the temporal growth prediction capabilities because the plant structure cannot change drastically, e.g., no completely new leaves can develop. Since this thesis implements larger growth prediction steps and plant structures change more in real field environments, it does not use plant shape prior. For evaluation, a coverage score (CS) is used, which compares the Intersection over Union (IoU) between generated and reference segmentations. The PLA evaluation of this thesis is closely related, but here, only the overall segmentation size is considered,

not the positioning of the segmentations via IoU, since structural differences in long-term predictions are to be expected and not necessarily an indication of poor model outcomes.

Yasrab et al. [97] focus on direct segmentation generation of future root and shoot systems of Arabidopsis and Komatsuna based on a time series of past images. For this, they use FutureGAN, a progressive growing GAN [98] coming from the video frame prediction domain, and achieve appealing results. As with previously mentioned models, their dependence on equidistant image time series is a limitation, as is the generation of only segmentation images, which cannot be used as artificial sensor data. Due to significant differences in the bit depth, the generation of segmentations is much less complex than RGB imagery.

In the work of Foerster et al. [93], GANs are used to predict powdery mildew spread on leaves using hyperspectral images, allowing prediction several days into the future. While in the aforementioned studies, only the relative growth prediction step is defined by model design, so the interval distance of equally spaced input sequence equals the prediction distance, Foerster et al. integrate the day to be generated after powdery mildew inoculation into the model as absolute time information. This is achieved by concatenating the time directly to the channel dimension of the images, which stabilizes the training at the same time. In this thesis, both approaches of implicit time integration by data pairing of different growth stages in Chap. 5 and explicit time integration by encoding the target growth stage in 7 and 6 are used.

To summarize, most methods are based on conditional GANs, whereby different architectures are implemented. The limitations are mainly in the integration of further conditions (except shape prior), the flexibility in the number and interval distance of the input images, and the prediction distance. Thus, long-term predictions with strongly changed plant morphology were not investigated. Furthermore, the utilization of image data from real field conditions and a calculation or visualization of the predicted uncertainty is under-explored. While this section provides a broad overview of related work on image-based modeling of crop growth, the following chapters focus on improving different aspects of prediction. Thus, integrated into the introduction of each chapter are specific, methodologically oriented state-of-the-art sections that include image-to-image translation (Sec. 5.1), data imputation and extrapolation for time series (Sec. 6.1), and combining images with multiple other conditions of different type (Sec. 7.1).

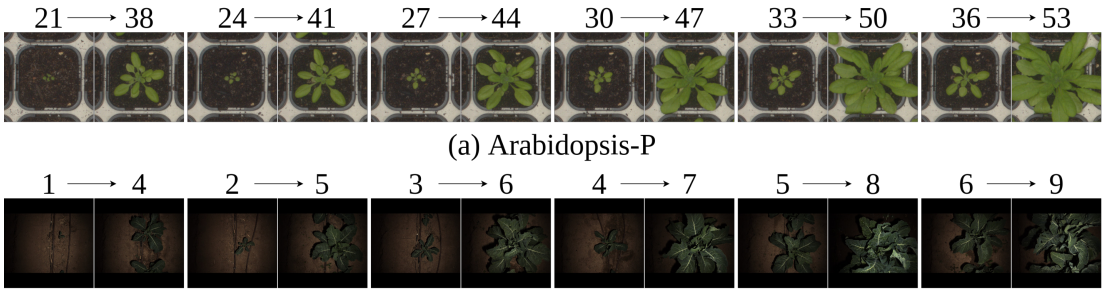
Chapter 4

Data

The studies conducted in this thesis are based on four different plant datasets, namely Arabidopsis (Arabidopsis-P und Arabidopsis-S), Brassica, GrowliFlower, and MixedCrop (Mixed-CKA, Mixed-WG). All of them show RGB images of different growth stages of their respective plants, from sowing or planting to harvesting or removal, either in a paired (Fig. 4.1) or in a sequential (Fig. 4.2) way. The acquisition time is available for each image and is given, depending on the plant species, relative to sowing in Days After Sowing (DAS) or relative to planting in Days After Planting (DAP) or in Weeks After Planting (WAP). One aim of this thesis is to show that the image-based CGMs apply to datasets with a wide range of different properties (Tab. 4.3). They differ technically in terms of overall size, image formats, and observation intervals during the vegetation period, but also semantically in essential properties. Different plant species, growing environments, and heterogeneities of plants are represented, as well as different resolutions due to varying cameras and image acquisition devices from ground robots to robotic measuring arms and UAVs.

4.1 Requirements

Certain minimum requirements must be met for a dataset to be suitable for image-based crop growth modeling. In general, plants need to be observed at different points in time over the growing period, and their age must be determinable - from the difference between the image acquisition time (usually contained in the image metadata) and the time of sowing or planting (often not available in plant datasets). Additional requirements apply to the observation process itself: All images in a dataset need the same spatial resolution referred to as GSD and a unified perspective on the plants. Preferably, a bird' s-eye view is used, in which plants can be assessed quickly and efficiently using remote sensing. Ideally, the same measurement setups should be used over time, and the lighting conditions



(a) Arabidopsis-P

(b) Brassica

Figure 4.1: Overview of paired datasets. The pairs always consist of an early growth stage in domain A (left) and an advanced growth stage in domain B (right), where the relative growth stage difference is always 17 d for (a) Arabidopsis-P and 3 w for (b) Brassica. The number above the images indicates the growth stage for (a) in Days After Sowing (DAS) and for (b) in Weeks After Planting (WAP).

should be constant, both of which are major challenges for observations in real field environments: the UAV flight height or the technical equipment can vary with changing demands from different drone imagery users and weather conditions strongly influence the lighting conditions. So, in real environments, these two requirements (setup and lighting) are maintained as well as possible, e.g., by UAV overflights always at the same daytime.

If Image Generation Models (IGMs) are applied in which one or more images are set as a condition, the input and the reference output need to be spatially aligned. Alignment means that a dataset shows images with the same section of a plant or the same field region over time. This is one of the most important/critical requirements, as the accuracy of the alignment depends on many aspects, such as the repeatability of the measurement configuration and, under real conditions, the accuracy of the geo-reference. All requirements taken together significantly limit the number of suitable datasets, especially under real field conditions, so datasets that fulfill the requirements with some restrictions are also permitted. A fulfillment overview for each dataset can be found in Tab. 4.3.

4.2 Datasets

This section provides detailed information on each image dataset used to build IGMs and corresponding information about treatments and process-based model results, which are used as conditions. Label data relevant to deriving plant traits from (generated) images in a supervised way within Growth Estimation Models (GEMs) are also addressed. The different GEMs used are described in Sec. 2.5.2. Despite partly higher resolution original images, whose pixel dimension and spatial resolution are mentioned below, all images in this work are resized to 256 × 256 px. We consider this size the optimal trade-off between the loss of

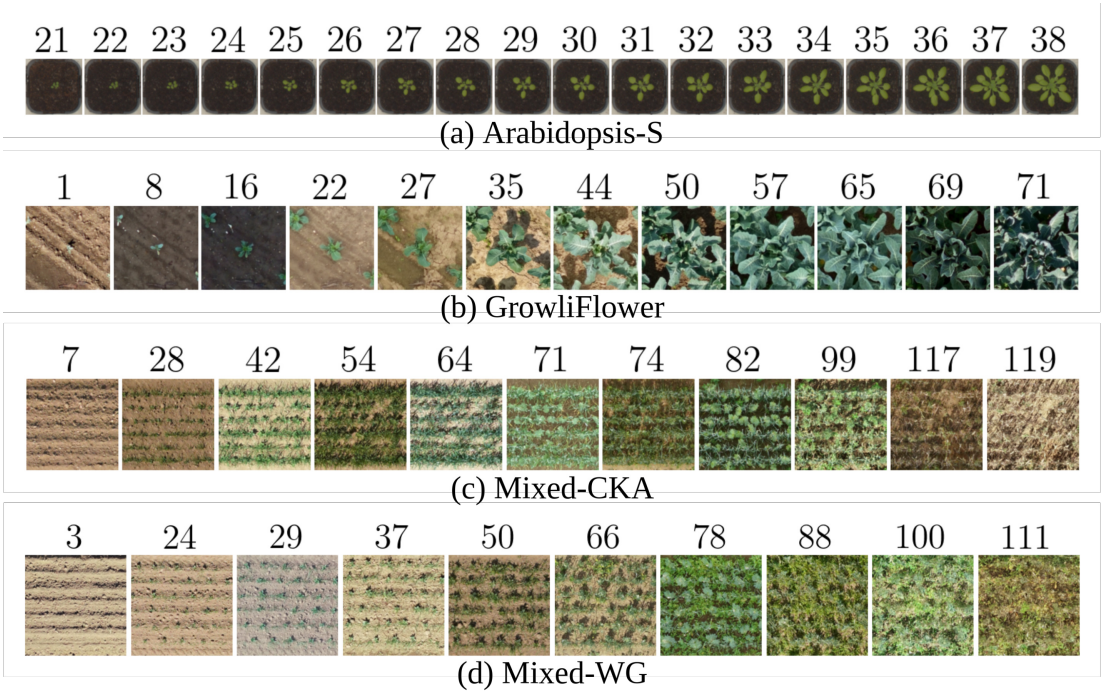


Figure 4.2: Overview of sequential datasets. Temporal development of one plant resp. field patch from each of the datasets (a) Arabidopsis-S, (b) GrowliFlower, (c) Mixed-CKA, and (d) Mixed-WG. The number above the images indicates the growth stage for (a), (c), and (d) in Days After Sowing (DAS) and for (b) in Days After Planting (DAP).

details in plant structures and efficient runtimes of the experiments. The experiments discuss to what extent the models can be transferred to smaller and larger images and what changes to the models may be necessary.

4.2.1 Arabidopsis

The two Arabidopsis datasets, a paired variant Arabidopsis-P and a sequential variant Arabidopsis-S, are created from the Aberystwyth leaf evaluation dataset [99]. It includes 80 different Arabidopsis (*Arabidopsis thaliana*) plants, which were all grown under the same treatment. They were recorded on 4 trays (tray031-034) of 20 plants each over a 35 d period from day 21 to day 55 after sowing using an IDS UI-5480SE camera (Tamron 8 mm f1.4 lens, 5 MP). Each tray image originally has a dimension of 1920 × 2560 px with a GSD of 0.15 mm. Up to four images were taken per hour, which led to 1676 observation times in total. The camera was mounted on a robotic arm in a controlled laboratory environment, ensuring high-accurate image alignment. For developing CGMs on individual plants, 20 plant-centered images are clipped out of each tray at each time point. Both the number of observation times used and the image clipping size are different for Arabidopsis-P and Arabidopsis-S, which affects the total number of images in the datasets. Segmentation labels on the leaf level are provided for

two trays and selected growth stages.

Arabidopsis-P

Arabidopsis-P is designed as a paired image dataset $\mathcal{P}$, where each image pair shows spatially aligned plants with a time difference of 17 days. Within each pair of images, domain A is the earlier growth stage, and domain B is the more advanced growth stage 17 days in the future. As plant leaf angles change over the course of the day, only plants recorded at the same time of day (± 15 min) are paired. While the relative time difference is consistent over the entire dataset, the absolute times of domains A and B are inconsistent. Domain A contains the growth stages from day 21 to 38 after sowing, while Domain B contains the growth stages from day 38 to 55 after sowing, resulting in a total of 10 325 pairs. Limiting the domains to only one point in time would result in a maximum of 80 image pairs due to the low number of different plants, which is comparatively small for training a deep generative model. Early growth steps are slightly over-represented in the dataset compared to later ones because half of the plants were removed from the trays before the observation ended, and in this case, no more pairs can be formed (the plant in domain B is missing). The clipping size for each plant-centered image is based on the time of plant expansion before there is a strong overlap with neighboring plants, which clearly exceeds the edges of the pot. However, especially in the later growth stages, neighboring plants can protrude into an image clipping and potentially overlap with the observed plant at the edges of the images. After resizing the images to 256×256 px, this results in a GSD of 0.32 mm. A spatial train-test split of the dataset is implemented: Plants from three trays, tray031, tray032, and tray033, are used for training, and plants from tray034 are used for testing.

Arabidopsis-S

Arabidospis-S is a sequential dataset $\mathcal{S}$ with a uniform sequence length of 850 images per sequence for the training and validation plants and 849 for test plants. In contrast to Arabidopsis-P, all tray images are corrected for barrel distortions with a provided calibration script [99] and then manually clipped to plant-centered images directly at the edges of the pots. After resizing to 256×256 px, this leads to a slightly higher GSD of 0.23 mm. This dataset focuses on images from 18 days of early developmental stages of *A. thaliana* from day 21 to day 38 after sowing since more advanced growth stages exceed the pot and thus the clipped image. Any plants that were already removed from the experiment before day 38 are excluded, leaving 64 plants. At the cost of a shorter observation period, which is reduced by half compared to Arabidopsis-P, there are benefits of a more

homogeneous dataset, with almost consistent length, no overlap in late growth stages due to neighboring plants, and no under-representation of advanced growth stages. Please note that the number of images per sequence clearly exceeds the duration of the observation period in unit [d] because not only one image per day was taken, but up to four per hour. There is a spatial train-val-test split: Two and a half trays were used for training (tray031, tray032, and half of tray033), half for validation (the other half of tray033), and one for testing (tray034).

4.2.2 Brassica

In the Brassica project [100], cauliflower and broccoli were observed in the field for a 10-week growing season under different treatments, whereby this work focuses on the 288 observed cauliflower heads (*Brassica oleracea var. botrytis*). The experimental site was in Australia at Lansdown Farm, about 70 km southwest of Sydney, transitioning between subtropical and temperate climates. The cauliflower heads are divided in the field into areas of four different treatments. They are arranged in two beds (namely, beds 01 and 03), each separated into four equally sized subareas with different irrigation and fertilization. The subareas are indicated with {`i+f+`, `i+f-`, `i-f+`, `i-f-`}, where `i` denotes irrigation, `f` fertilization and `+`/`-` mean sufficient or insufficient conditions, respectively.

Multi-modal images (RGB, hyperspectral, thermal) were acquired with an autonomous ground-based kinematic multi-sensor system, of which the RGB images from two Grasshopper3 12MP GS3-U3-120S6C-C cameras in a stereo setup are used. Due to the stereo setting, the plants are usually not centered in the images but are slightly offset to the right or left. Weekly image acquisition took place during movement at a speed of 0.1 m/s and an acquisition rate of 0.5 Hz in the first part of the growth period (weeks 1-4) and 1 Hz in the second half (weeks 5-10), resulting in 4 or 8 images of the same plant at maximum. Due to the continuous recording, many images are taken between two plants. These are removed from the dataset since bare soil can be seen mainly. Images from the first to ninth week after sowing are used; the tenth week is not included because most cauliflower plants at this growth stage already exceed the image dimension. Initially the images have a non-square shape of 706 × 1060 px with a GSD of 1.14 mm, whereby resizing to 170 × 256 px results in a GSD of 4.84 mm and zero-padding of 43 px rows applied at the top and bottom ensures square image sizes of 256 × 256 px.

The images' geo-reference, which is used for image alignment, is achieved via the localization units (IMU and GNSS) integrated into the ground robot and enables monitoring of the same plant over time. However, it was not successful to hold the same line on different measurement days within a few millimeters. Therefore, two images are considered spatially aligned if their geo-positions are

less than a threshold of 2 cm apart. Since it is more difficult to maintain an alignment the more time points are taken into account, the dataset is not designed as a sequential but as a paired dataset $\mathcal{P}$ with a difference in growth of 3 weeks. So images of week 1 are paired with images of week 4, week 2 with 5, up to pairing of week 6 with 9. Similar to Arabidopsis-P, both domains of the pairs contain several growth stages: Domain A of the image pairs ranges from week 1 to week 6 and domain B from weeks 4 to 9, which leads to duplicate uses of images in the overlapping weeks 4, 5, and 6. After removing images in spaces between plants and images in which the cauliflower has already been harvested or removed in another way, and after applying the alignment threshold, 6 658 images (3 329 pairs) remain of the original 21 928 images. However, image pairs where plants are partly visible in both domains by at least 50 % are not removed as they contribute to the diversity of the dataset.

The dataset is divided into spatially disjoint train-test parts: for training, the plants from bed 01 are used, and for testing, the plants from bed 03. Since a distinction between different treatments is not made during training, all the cauliflower plants from the entire bed 01 are used. For testing, the 3-week aligned image pairs of bed 03 are separated into the field treatments {`i+f+`, `i+f-`, `i-f+`, `i-f-`}, which is shown in Tab. 4.1.

Table 4.1: Number of 3-week aligned cauliflower image pairs divided in bed 01 (train) and bed 03 (test) of the Brassica dataset. For training, the whole bed 01 is used. The test data is divided into the treatment regions {`i+f+`, `i+f-`, `i-f+`, `i-f-`}. A sensor outage in week 7 results for step $4 \rightarrow 7$ in only pairs for the `i+f-` region of bed 03.

Week	Train: Bed 01	Test: Bed 03				
	$\sum$	`i+f+`	`i+f-`	`i-f+`	`i-f-`	$\sum$
$1 \rightarrow 4$	124	43	57	44	46	190
$2 \rightarrow 5$	322	110	82	94	90	376
$3 \rightarrow 6$	198	72	70	72	80	294
$4 \rightarrow 7$	0	0	86	0	0	86
$5 \rightarrow 8$	407	166	100	152	104	522
$6 \rightarrow 9$	270	148	118	124	150	540

A few images of various growth stages were handpicked to be annotated for plant-level instance segmentation. 35 images are labeled in total, 25 of which are training images and 10 of which are test images. These labels are used to set up an instance segmentation model that derives the overall plant size and the plant's location in the image from RGB input. Thereby, the plant size is determined by multiplying the Projected Leaf Area (PLA) in the unit pixel by the GSD.

There were three measurement challenges affecting the images in the dataset.

First, ensuring adequate and constant lighting conditions. Images are taken underneath the robot and are generally not particularly bright. The artificial light source causes leaves to be brighter in later growth stages because they are grown closer to the light. Second, due to a sensor outage in week 7, there are no image pairs for step $4 \rightarrow 7$ in the training set and only fewer pairs in section `i+f-` of the test set. Third, in the imagery of week 8, small dot-like oversaturations with maximum reflectance values occur in all color channels, resulting in white stains.

4.2.3 GrowliFlower

The GrowliFlower dataset [52] is a collection of multiple image data of cauliflower (*Brassica oleracea var. botrytis*) from two different fields in Bornheim, Rhein-Sieg Kreis, Germany. From this collection, unlabeled RGB image time-series (GrowliFlowerT, field 2) are used as sequential data $\mathcal{S}$ for training the IGM, showing 8 522 cauliflower plants in the growing period from June to September 2021. In this thesis, a total of 102 264 images taken on 12 measurement days within a period of 71 days from June to August are used, whereby images after harvest are excluded. All plants were grown under uniform external conditions, although soil inhomogeneities cannot be avoided. The original images stem from orthophotos taken by an UAV equipped with a Sony Alpha 7R III camera (Zeiss/Batis 2.0 lens, 47.4 MP). Aligned plant-centered clippings - enabled by the orthophotos geo-reference - are taken from these orthophotos, which have a dimension of 256×256 px and a GSD of 3.10 mm. Finally, there is not only one plant per image, but (parts from) up to four heads are visible at the image edges and overlap in advanced growth stages. The whole dataset is divided into sets of approx. 77 % training and 11.5 % validation and test each, whereby a spatial separation ensures that no center plant is visible in multiple sets.

A total of 2 197 labeled images of all growth stages of cauliflower heads of another field (GrowliFlowerL, field 1) are provided, which are used to train a GEM. The labels are made for instance segmentation of leaves and whole plants, whereby this work is focused on the latter. The resolution of the labeled images is slightly higher than that of the image time series with a GSD of 1.65 mm on an image size of 448×368 px.

4.2.4 MixedCrop

The MixedCrop data are from a 2020 and 2021 PhenoRob crop mixture experiment described in detail by Paul et al. [53]. Two different cultivars of faba bean (FB, *Vicia faba*) and twelve different entries of spring wheat (SW, *Triticum aestivum*) were sown in mixtures of a 1:1 ratio, which means 50 % of seeds of each species from the respective monoculture as well as in monocultures. An overview

of the faba bean cultivars and spring wheat entities used in the MixedCrop experiment is given in Tab. 4.2. Coupled with two different seeding densities i.e. low (L) 80% and high (H) 120% of the recommended sole crop densities (400 seeds m^{-2} for SW and 45 seeds m^{-2} for FB), this results in $(2 \cdot 12 + 2 + 12) \cdot 2 = 76$ different treatments, which were replicated four times, or, in case of the faba bean monocultures, eight times, resulting in a total of 320 different plots of size $10\,m \times 1.5\,m$. The same setup was applied to field experiments in 2020 and 2021 at two different research sites of the University Bonn, which are both located in the Rhein-Sieg-Kreis, Germany. The first, Mixed-CKA, is at Campus Klein-Altendorf (CKA, near Rheinbach), and the second is at Wiesengut (WG, near Hennef). Both experimental sites are located about 30 km apart and have significantly different growing conditions because Mixed-CKA is managed conventionally and Mixed-WG organically. Along with an UAV image campaign, a variety of field data were collected multiple times during the growing period, including weather, soil, and nutrient parameters as well as manual height and biomass measurements [53], [101]. In this thesis, the focus for both sites is on the experiments of the year 2020.

The RGB-image acquisition was conducted 11 times for Mixed-CKA[1] and 10 times for Mixed-WG[2] by UAV equipped with an FC6310 camera (1″ CMOS 8.8 mm, 20 MP). The 320 field plots are positional-aligned clippings from the geo-referenced orthophotos before being horizontally rotated and plot-centered clipped into seven non-overlapping and square image patches. Five of these patches are used for training, and one each is used for validation and testing so that all sets keep all the different treatments. Each patch forms a sequential sample over time, making both Mixed-CKA and Mixed-WG sequential data sets $\mathcal{S}$. While the original orthophotos have a GSD of 3 mm, the image patches resized to 256×256 px have a resolution of 5.67 mm. Due to orthophoto corruptions and destructive field measurements, some sections were manually removed, resulting in a final number of 21 371 images for Mixed-CKA and 18 800 images for Mixed-WG. For Mixed-WG, a significant spatial alignment error was noticed by visual inspection, which is up to 10 cm, but inconsistent (offset in different directions) across the images and, therefore, difficult to filter out. Since 10 cm corresponds approximately to the spatial extent of a faba bean plant at 20 days after sowing (DAS), the offset is well visible in the early growth stages.

[1]Mixed-CKA field patches are available at `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/751c10c4-b6dc-4bcc-bc8c-c0fc5920887a`

[2]Mixed-WG field patches are available at `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/d9d0434f-7864-435e-9c75-56102d9332cb`

Table 4.2: Notation overview of species faba bean (FB) with cultivars A and B and spring wheat (SW) with cultivars 1-10 and two additional mixed groups used in this work.

FB (Faba bean)	A	Mallory
	B	Fanfare
SW (spring wheat)	1	Lennox
	2	Anabel
	3	Saludo
	4	Jasmund
	5	Sorbas
	6	Quintus
	7	KWS Starlight
	8	Chamsin
	9	Sonett
	10	SU Ahab
	11	Mix-Group 1
	12	Mix-Group 2

Biomass simulation using process-based modeling of crop mixtures

Labels for all images, which consist of the simulated dried biomass according to growth stage and treatment, are calculated using a PBM for crop mixtures. The process-based crop growth simulations were conducted in the modeling platform SIMPLACE (Scientific Impact Assessment and Modeling Platform for Advanced Crop Ecosystem Management) [70]. Different sub-models in the SIMPLACE framework, called "SimComponents", were combined, namely LINTULPhenology, LINTUL5NPKDemand, SlimNitrogen, LINTUL5Biomass, SlimRoots, and SlimWater, among others. An overview of key SimComponents[3] is described in Seidel et al. [102]. Specifically, the biomass per species was calculated by SimComponent LINTUL5Biomass, which considers the effects of water and nitrogen limitation on biomass increment. Further mixture effects are taken into account within the SIMPLACE framework by simulating the splitting of solar radiation according to the competition of the two species as well as the water and nitrogen uptake of two crop species planted in a mixture. The model was calibrated and tested in three environments (CKA 2020, 2021, and WG 2020) based on the data collected from the crops cultivated solely and evaluated based on the data in the mixture treatments.

[3]Information about SIMPLACE components: `www.simplace.net/doc/simplace_modules`

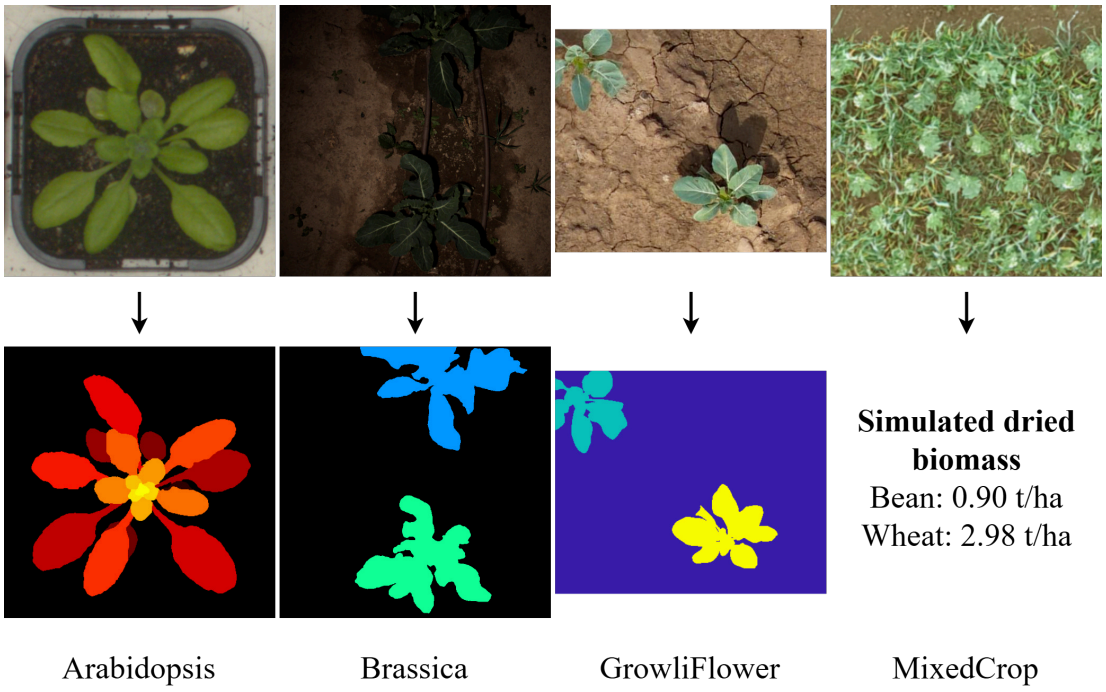


Figure 4.3: Overview of dataset labels. Instance segmentation for Arabidopsis, Brassica, and GrowliFlower and process-based simulated dried biomass values for MixedCrop.

4.3 Comparison

A comparison between the datasets reveals many significant differences, as Tab. 4.3 shows. When comparing the paired datasets, the challenges of data collection under real field conditions become apparent. Brassica has significantly fewer images, fewer measurement times, and a lower spatial resolution than Arabidopsis-P. In addition, there are difficulties with satisfactory exposure and spatial alignment with the field robot, whereas Arabidopsis-P represents a very homogeneous dataset due to the laboratory conditions. However, significantly fewer plants were observed with Arabidopsis-P and only under a single treatment (compared to Brasscia: 4), which limits the diversity of the dataset and its generalizability to Arabidopsis plants outside the dataset.

There are two datasets containing cauliflower images, Brassica and GrowliFlower, with GrowliFlower being about 15 times larger. Regarding spatial alignment, UAV has advantages over Unmanned Ground Vehicle (UGV): To clip the same positions over time and thus generate sequential data, precise coordinates of each field point in a world coordinate system are required. This can be realized for UAV with the recording of ground control points and the calculation of an orthomosaic via bundle block adjustment. Meanwhile, this is not an option for UGVs due to the smaller field of view and the need for additional positioning sensors. In turn, the UGV succeeds in providing weak but constant light conditions

through artificial light, while the illumination of orthophotos depends on solar radiation, cloud cover, and the current properties of the soil (composition, structure, and moisture) and therefore varies strongly from one measurement time to the next.

A comparison of the sequential datasets shows that very different types of plants are examined: stand-alone single plants (without overlap), cabbages (with and without overlap), and crops with strong overlaps even at early growth stages. Not only is the sequence length different, but particularly with the sampling frequency, gaps of different sizes between the recording times are noticeable. While there are several daily observations of only 64 plants for Arabidopsis-S, there are irregular observations and gaps of several days for GrowliFlower and MixedCrop, but significantly more different cabbages or crops. Considering the overall size, which decreases from GrowliFlower (~100 000 images) to Arabidopsis (~50 000) to MixedCrop (~20 000), the latter is particularly challenging as there exist 76 different treatments in addition to the small dataset size.

There are also differences in the labels between the datasets, as seen in Fig. 4.3, which influences the creation of GEM. While simulated biomass data are available for MixedCrop and suitable for image regression tasks, pixel-wise plant instance segmentations are available for all other datasets at different scales - leaf instances or whole plant instances. In this work, only the projected leaf area per plant or image is evaluated so that the instance segmentation of whole plants represents a sufficient level of detail.

Table 4.3: Overview of all dataset properties. The upper block gives general specifications about the datasets, the middle block indicates the number of images available for the image generation model (IGM), and the bottom block displays the number of labeled images available for the resp. growth estimation models (GEM). [1]Ranking of requirement fulfillment (a) resolution, (b) perspective, (c) setup, (d) lighting, and (e) alignment (details in Sec. 4.1) into good (✓), medium (~), and bad (×). *No fixed validation set; instead, a proportion of random samples from the training set are used for validation.

	Arabidopsis		Brassica	GrowliFlower	MixedCrop	
	Arabidopsis-P	Arabidopsis-S			Mixed-CKA	Mixed-WG
dataset type	paired	sequential	paired	sequential	sequential	sequential
# plants	80	64	288	8 522	uncounted	uncounted
# treatments	1	1	4	1	76	76
# observation times	35	850	9	12	11	10
observation period [d]	35	18	63	71	113	109
image size [px]	256×256	256×256	170×256	256×256	256×256	256×256
GSD [mm]	0.32	0.23	4.84	3.10	5.67	5.67
requirements (a|b|c|d|e)[1] met?	✓|✓|✓|✓|✓	✓|✓|✓|✓|✓	✓|~|✓|~|~	✓|✓|✓|×|✓	✓|✓|~|×|✓	✓|✓|~|×|~
IGM: # images	20 650	54 384	6 658	102 264	21 371	18 800
IGM: # train images	15 236	34 000	2 642	78 864	15 017	13 154
IGM: # val images	*	6 800	*	11 748	3 177	2 823
IGM: # test images	5 414	13 584	4 016	11 652	3 177	2 823
IGM: # sequences or pairs	10 325	64	3 329	8 522	2226	2212
IGM: ∅ images per sequence	-	849.75	-	12	9.60	8.50
GEM: # labeled images	1 100	808	35	2 197	21 371	18 800
GEM: # train images	850	512	25	1 541	15 017	13 154
GEM: # val images	*	148	*	326	3 177	2 823
GEM: # test images	250	148	10	330	3 177	2 823

Chapter 5

Paired image-to-image translation

This chapter is about predicting plant growth by generating a realistic future image from an input image with a predefined growth stage offset. Since no other conditions besides the image of the early growth stage are included, the process is also called paired image-to-image translation: a plant's image of an early growth stage gets transformed into an image of a future growth stage. The aim is to develop data-driven CGMs for the paired image datasets Arabidopsis-P and Brassica that generate realistic and reasonable images with a specific fixed time interval relative to the input time. Realistic means that the appearance of the generated plant images is not distinguishable from reference plant images at the same growth stage. Reasonable means that plant traits derived from the generated images are in line with traits assessed of reference plants, which is analyzed in a comprehensive evaluation.

As growth prediction steps, we have chosen long-term predictions, specifically time intervals of 17 d for Arabidopsis-P and 3 w for Brassica, so that structural changes are to be expected on the images instead of just pixel-by-pixel color changes. In the Brassica dataset, the plants were actively exposed to different treatments by fertilization and irrigation. However, these management decisions are not explicitly integrated into CGM. For this reason, it is being investigated whether these growth influencing factors, which are only noticeable by minimal differences in the images of the early growth stages, can be captured and reasonably processed by the CGM.

For this purpose, we introduce a CGM workflow based on three steps, as depicted in Fig. 5.1. First, we train a CGAN, which is based on the pix2pix model by Isola et al. [103] in a data-driven way with the use of image pairs showing an early and late growth stage. Second, we use this model's generator to generate predictions for new images of the early growth stage. Third, these

are evaluated both in the image space by instance segmentation and in latent space using FID.

The main contributions of this chapter can be summarized as follows:

- Generation of realistic and reasonable long-term predictions for plant growth via image-to-image translation as CGM, where the input and target image are structurally significantly different.

- Investigation of how treatment differences in the input images affect the generated images without explicitly incorporating them as growth influencing factors into the CGM.

- Comprehensive analysis of the generated images using different FID variants and the deviation of the PLA derived from instance segmentation between reference and generated plants.

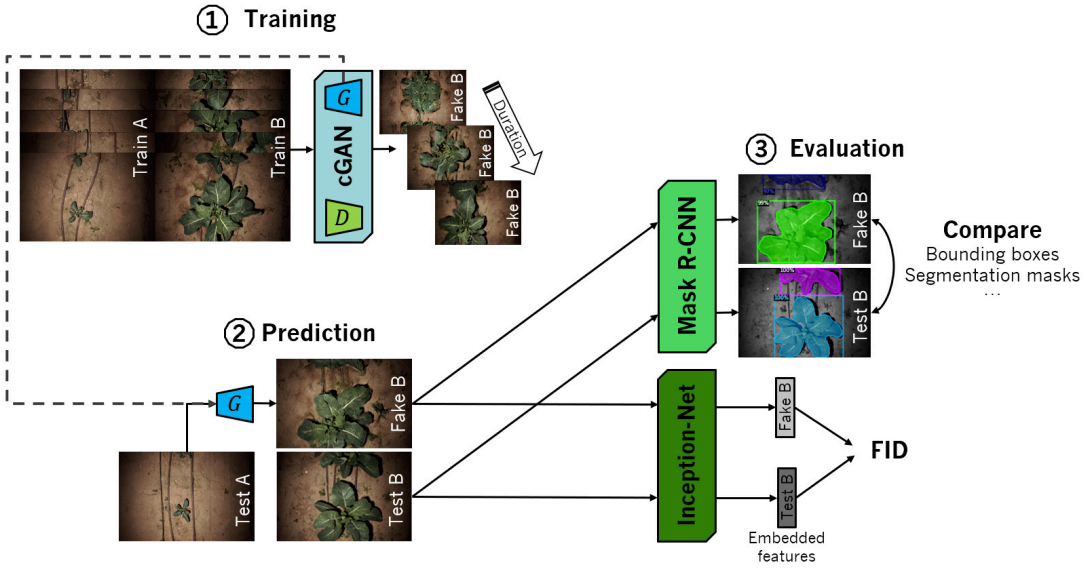


Figure 5.1: Paired image-to-image translation pipeline: First, a conditional GAN is trained on training pairs of domains A and B. Through adversarial training of discriminator ($\mathcal{D}$) and generator ($\mathcal{G}$), the generated images become more realistic with each epoch. Second, the generator is used to generate predictions from the test input. The third step of evaluation is divided into instance segmentation using Mask R-CNN and FID calculation. Instances are calculated on generated images as well as on reference images with the use of a Mask R-CNN model. The comparison of the instance parameters (bounding box, area of the segmentation mask) allows a statement about the quality of the generator. The FID score provides an additional objective measure of the quality of the generated plant images.

5.1 State of the art

The generation of artificial images with GANs has recently found increased attention in agriculture and plant science. The underlying challenge of the image-to-image translation task using GANs is to obtain a translation between two domains, A and B, where a so-called domain is a set of data samples such as images whose distribution is implicitly determined by the GAN. The various GAN approaches presented in the literature differ in how domains A and B are chosen. In our work, we will refer to the early plant stage as the source distribution, denoted as domain A, and the advanced plant stage as the target distribution, denoted as domain B.

A commonly used type of GANs for agricultural applications are Cycle-consistent GANs (CycleGANs), e.g., for the detection and discrimination of plant diseases and the estimation of their future spread on its leaves [88], [93], [104]. Other applications include the translation of real images (domain A) into outputs (domain B) that directly contain interpretations of the data, such as semantic segmentations [85]. Furthermore, outputs can also be products such as vegetation indices like the normalized difference vegetation indices [105]. The image-to-image translation is also suitable for data augmentation and up-scaling of plant imagery, which produces new higher-resolution images from low-resolution ones and thus enables the analysis of plant traits in a more detailed way [106], [107].

CycleGANs are particularly suitable, if necessary, to translate in both directions, from domain A to domain B and vice versa. Thereby, they do not require aligned image pairs, which means that for an image from domain A, there does not have to exist a corresponding image of the same plant in domain B [108]. The ability to use non-aligned data is essential for many applications that have sufficient training data from both domains, e.g., leaves with and without disease, but which have only a few image pairs [93].

In agriculture, aligned temporal image pairs are becoming widely available due to geo-referenced orthophotos or by using kinematic multi-sensor systems, which help to position sensor imaging data, for example, by GPS. In order to exploit this specific data characteristic, CGANs can be used to learn a powerful generator based on a given set of input and output pairs [103]. These networks show for various application areas that they can achieve good results in the field of domain adaptation [109]–[111], but they have rarely been used for crop growth modeling so far [94], [97]. To our knowledge, there is no related work yet on the utility of agricultural data pairs in CGANs where the domains differ significantly in time, as is done in this work.

A few works have already successfully used plant data in CGANs, like artificial targeted plant generation, for the aim of data augmentation, which is useful if only a limited amount of training data is available [112]. For this purpose, Zhu et

al. use segmentation masks of plants on the input side of a CGAN to synthesize new real-looking plant images on the output side [86]. In addition to pairs of images of the same size, other corresponding pairs of conditions and outputs can be used, e.g., when generating images depending on scalars or one class label. For instance, Giuffrida et al. generate images with plants of different sizes, where the number of leaves is introduced as a scalar condition attached to a noise vector [113]. The resulting generated images can be used to augment data, correct imbalances, and generate adequate samples in the training set.

Apart from CGANs, diffusion models recently entered the area of image-to-image translation, enabling, for example, high-quality colorization and inpainting [114]. Within the same temporal domain, they have already been used for plant data, e.g. to translate healthy leaves into diseased leaves for data augmentation [115]. An impression of the potential of conditional latent diffusion models is given in Sec. 8.3.1.

5.2 Methods

5.2.1 Conditional GAN for image-to-image translation

The CGAN model for image-to-image translation is inspired by the Pix2Pix model of Isola et al. [103] with adjustments made to hyperparameters, as described in more detail below. With this model, a mapping $\mathcal{G} : \{{}^{\mathrm{A}}\mathsf{X}, \epsilon\} \rightarrow {}^{\mathrm{B}}\mathsf{X}$ is implemented from images of domain A ($^{\mathrm{A}}\mathsf{X}$) and random noise ϵ to images of domain B ($^{\mathrm{B}}\mathsf{X}$). Since the mapping represents the transformation from one image (with noise) to another image, it is referred to as a translation in the computer vision context. The classic CGAN optimization is applied as described in Sec. 2.4.2, where the conditions y are exclusively represented by an image of domain A.

$$\begin{aligned}\mathcal{L}_{\mathrm{CGAN}}(\theta, \delta; {}^{\mathrm{A}}\mathsf{X}, {}^{\mathrm{B}}\mathsf{X}, \epsilon) = \mathbb{E}_{{}^{\mathrm{A}}\mathsf{X}, {}^{\mathrm{B}}\mathsf{X}}[\log(\mathcal{D}_\delta({}^{\mathrm{A}}\mathsf{X}, {}^{\mathrm{B}}\mathsf{X}))] + \\ \mathbb{E}_{{}^{\mathrm{A}}\mathsf{X}, \epsilon}[\log(1 - \mathcal{D}_\delta({}^{\mathrm{A}}\mathsf{X}, \mathcal{G}_\theta({}^{\mathrm{A}}\mathsf{X}, \epsilon)))]\end{aligned} \tag{5.1}$$

While the adversarial loss encourages the generation of diverse and realistic samples, an additional L1-loss is added to maintain a high degree of similarity to the target image in a pixel-wise manner [116]. Therefore, the L1-loss is also referred to as reconstruction loss, which only has an influence on the generator weights θ while discriminator weights δ remain unaffected.

$$\mathcal{L}_{\mathrm{L1}}(\theta; {}^{\mathrm{A}}\mathsf{X}, {}^{\mathrm{B}}\mathsf{X}, \epsilon) = \mathbb{E}_{{}^{\mathrm{A}}\mathsf{X}, {}^{\mathrm{B}}\mathsf{X}}\left[\| {}^{\mathrm{B}}\mathsf{X} - \mathcal{G}_\theta({}^{\mathrm{A}}\mathsf{X}, \epsilon)\|_1\right] \tag{5.2}$$

Combined, the two loss functions result in the total objective, with the hyperparameter λ controlling the weighting between the adversarial part and the reconstruction part.

$$\theta^*, \delta^* = \arg\min_\theta \arg\max_\delta \mathcal{L}_{\mathrm{CGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta) + \lambda \mathcal{L}_{\mathrm{L1}}(\mathcal{G}_\theta) \tag{5.3}$$

To ensure balanced adversarial training, backpropagation is performed alternately on the discriminator first and then on the generator.

An important difference to classical GAN is the way stochasticitiy is integrated. In this model it is not included as a Gaussian noise vector, but the stochastic component is realized by dropout layers, which are enabled both during training and inference. Previous studies have found that noise with classical GAN optimization is suppressed when it is processed together with an image in the input because the image represents a greater information content for the translation task [103].

Generator architecture

The generator $\mathcal{G}$ network is a U-Net [117] with skip-connections. Hereby, the input images are first processed in an encoder architecture until a bottleneck layer, to which a symmetrical decoder structure is attached. The encoder consists of 8 convolutional down-sampling blocks, each representing a sequence of a strided convolutional layer, LeakyReLU activation, and batch normalization. Accordingly, the decoder consists of 8 convolutional upsampling blocks, where the spatial upsampling in each block is performed by a strided transposed convolution layer followed by ReLU activation and batch normalization. After each upsampling, there is a dropout of 50 %, except for the innermost and outermost convolutional blocks. Skip connections are used to preserve significant features of earlier layers, such as edges indicating the orientation of leaves, the overall size of plants, or the background structure.

Discriminator architecture

The discriminator $\mathcal{D}$ is a convolutional PatchGAN architecture that classifies patches of the generated images into real and fake. In contrast to the classic discriminator, which outputs a classification score for the whole image, PatchGAN evaluates structures at the scale of local image patches. For this purpose, 3 strided convolution blocks are used first, followed by a non-strided convolution block, both of which increase the filter number piece by piece, and a final non-strided convolution layer to reduce the filter dimension to one. In each block, batch normalization and LeakyReLU activation are applied after convolution. This special discriminator architecture has the advantage of focusing on the high-frequency correctness of images. The low-frequency correctness is slightly neglected, but this is compensated by the simultaneous use of the L1-Loss. In this way, the images get a finer structure and a better texture [103].

5.2.2 Evaluation of generated images

Evaluation by instance segmentation

The first part of the evaluation focuses on the appearance of the plants. For this, manually derived segmentation masks of the real images are compared to estimated segmentation masks of the generated images by means of parameters such as extent and area. Since semantic segmentation, where each pixel is assigned a class, is insufficient due to limitations when plants overlap, we utilize Mask R-CNN to compute segmentation masks, which are semantic segmentations of each individual plant instance. The concept behind Mask R-CNN models is described in Sec. 2.5.2, and specifically, the Detectron2 framework [118] is used. The instance segmentation masks can be used to quantify various plant traits. The segmentation area is used to determine the plants' size, i.e., the number of pixels covering the plant, which can be converted to the PLA given the spatial resolution of the dataset (GSD). In addition to the segmentation masks, the estimated bounding boxes of the instances are used to determine the two traits: plant center and width. Here, the center position is defined as the center of the leaf extent, which approximates the plant's actual center. We focus on the width of the plants (inter-row) rather than the height (intra-row). For Arabidopsis-P, there is no significant difference, but for Brassica, the inter-row spacing of cauliflower in the field is higher, so plants are in width less affected by overlapping errors.

Evaluation by Fréchet inception distance

The second part of the evaluation focuses on the Fréchet Inception Distance (FID) [38], which calculates the distance between the real and the generated multivariate Gaussian image distribution, as described in Sec. 2.5.1. Unlike most related work, we analyze three different FID scores, which compare the similarity between the distributions of the generated, the test-reference, and the training-reference images:

- FID($\mathcal{N}_g, \mathcal{N}_r$): generated vs. test-reference
- FID($\mathcal{N}_g, \mathcal{N}_t$): generated vs. training-reference
- FID($\mathcal{N}_r, \mathcal{N}_t$): test-reference vs. training-reference

This provides insight into whether the model is actually capable of generating new images based on the test conditions or whether training images are being replicated. It is expected that FID($\mathcal{N}_g, \mathcal{N}_r$) is smaller than FID($\mathcal{N}_g, \mathcal{N}_t$), where not the same plants are compared with each other. That is because, in order to generate the images $\mathcal{N}_g$, the model receives as condition domain A from the test-reference $\mathcal{N}_r$ and not from the training-reference $\mathcal{N}_t$. FID($\mathcal{N}_r, \mathcal{N}_t$) represents

a control value in which only real image distributions are compared with each other. If the generated image quality is very good, $\mathrm{FID}(\mathcal{N}_g, \mathcal{N}_r)$ should also be smaller than $\mathrm{FID}(\mathcal{N}_r, \mathcal{N}_t)$, because the latter again compares different plants with each other. However, a very low value is to be expected for $\mathrm{FID}(\mathcal{N}_r, \mathcal{N}_t)$ if the test and training plants are similar. So, this control value also indicates the difficulty and complexity of the dataset.

5.3 Experiments and results

5.3.1 Experimental setup

In the experiments, we calculate a CGM for both paired image datasets Arabidopsis-P and Brassica. A growth prediction step of 17 d into the future is to be enabled for Arabidopsis-P and a step of 3 w for Brassica. It needs to be emphasized that the models should not be limited to a specific input time but should be able to simultaneously perform several prediction steps of the fixed length (17 d or 3 w) within the growth period. For instance, using the Arabidopsis-P, the same model should be able to predict the steps of 21 $\rightarrow$ 38 DAS, 22 $\rightarrow$ 39 DAS, up to 38 $\rightarrow$ 55 DAS. Accordingly, using Brassica, the predictions for the steps of 1 $\rightarrow$ 4 WAP, 2 $\rightarrow$ 5 WAP, up to 6 $\rightarrow$ 9 WAP are to be made with the same model.

Experimental goals

There are different expectations of the generated images in terms of realistic appearance, reliability, and reasonableness:

- Realistic appearance: The images as a whole should show details, minimal blurring, and no artifacts. Plants should look natural in terms of color, structure, texture, and size.
- Reliable generation: The model should have a high generalization ability, i.e., the generation should not only work for a part of the dataset but be robust to different growth stages, shapes of the plant, and background conditions.
- Reasonable output: The generated image should not be arbitrary but depend on the input image. If a model is trained on different field treatments, it should be able to predict different plant sizes accordingly.

It must be noted that we do not expect the generated image to contain every detail of the reference image. Especially the orientation and size of single leaves vary strongly between growth stages that are 17 d or 3 w apart and, therefore,

cannot be reconstructed. Rather, we expect to produce a plant similar to the reference in terms of overall size and position in the image, and thus, geo-referencing in the field to be within the accuracy of the image alignment. It is also expected that the highly accurate image alignment and almost continuous observation positively influence the results of Arabidopsis-P compared to Brassica.

Augmentation and hyperparameter

Since the generator is most efficient for square images, the rectangular Brassica images are provided with an equally large black margin on the top and bottom sides, as described in Sec. 4.2.2. Data augmentation consists of random cropping, vertical and horizontal mirroring, and rotations applied to both domains of the train and test image pairs. For Brassica, the rotation options are limited to 0 deg and 180 deg to maintain the geometry of the vertical alignment of the cauliflower rows in each image. The network architecture is maintained in its original state as presented in [103]; however, some hyperparameters are adjusted. The learning rate is set to $1e-4$, the loss weighting parameter λ to 100, and the batch size to 1. The number of epochs is 160 for Brassica and 40 for Arabidopsis-P, the second half of which has a linearly decaying learning rate.

Runtime

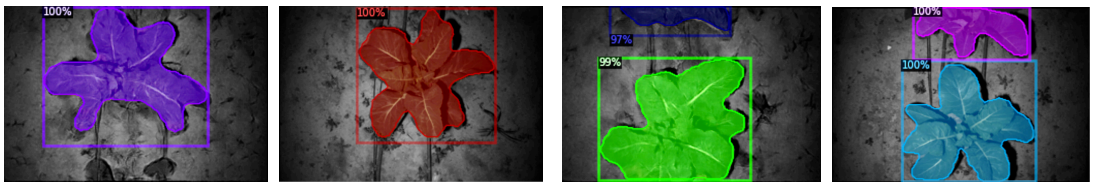
Following the pipeline in Fig. 5.1, the runtime for each step has to be considered separately. To train a suitable generator, the computing time is about 5 min per epoch for Arabidopsis-P and about 3.5 min per epoch for Brassica, whereby this time is largely determined by the dataset size and the number of augmentations. The convergence time of GAN training depends mainly on the diversity of the image distribution, which is higher for Brassica than for Arabidopsis-P, requiring more epochs overall (160 instead of 40). The two following inference steps, namely, applying the generator to predict future images and calculating the projected leaf area using the pre-trained Mask R-CNN, are real-time capable.

5.3.2 Accuracy assessment of instance segmentation

In order to compare the instance segmentation between the generated and real plants, it is first necessary to analyze how accurate the instance segmentation is with regard to labeled reference data. The Detectron2 framework is pre-trained on everyday objects of the large-scale COCO dataset [119] and fine-tuned on labeled images from the respective datasets. The train and test data for fine-tuning include images from all growth stages. For instance segmentation of *A. thaliana*, the Mask R-CNN model is fine-tuned on about 850 images and evaluated on about 250 images. The instance segmentation of cauliflower in the Brassica dataset is

fine-tuned using 25 images and evaluated using 10 images. The amount of training data is significantly lower for Brassica because manual labeling was required, as described in Sec. 4.2.2. Although the Brassica dataset has a small amount of training data for fine-tuning, it is sufficient because basic features are already learned in the comprehensive pre-training. A separate instance segmentation model is trained for each dataset, wherein both datasets we restrict ourselves to two classes: plant and background.

For both datasets, high-quality bounding boxes with an average precision $>$ 75 % and semantic instance segmentation masks with an average precision $>$ 70 % are estimated. In all experiments, the same instance segmentation model is applied to the reference and the generated images, as visualized in two examples in Fig. 5.2. Different image qualities can also lead to different instance segmentation and thus to uncertainties, even though the leaf areas of two images are actually identical: If the generated images have quality deviations that do not occur in the real images, e.g., blurring, instances may not be correctly recognized on the generated images. Therefore, the basis for the evaluation by means of instance segmentation is a good image quality that has already been qualitatively assessed.



generated A | reference A | generated B | reference B

Figure 5.2: Two examples, A and B, of instance segmentation using Mask R-CNN performed on two pairs of generated and associated reference images of cauliflower growth week 5. Colors are chosen randomly and have no meaning. The classification certainty is indicated in the corners of the boxes.

5.3.3 Results of qualitative evaluation

Visual analysis of generated Arabidopsis-P images

Fig. 5.3 shows 3 examples of visual results of temporal prediction in the Arabidopsis-P dataset. The upper row shows the prediction from 23 $\rightarrow$ 40 DAS, the middle row from 30 $\rightarrow$ 47 DAS, and the lower row from 37 $\rightarrow$ 54 DAS. The prediction is successful in both early and late epochs since the generated images are highly similar to the reference images, both in terms of the extent and the number of leaves. There are only a few details that reveal the artificiality of the generated plants. For instance, in the generated image on day 40 (upper row, center), there are two leaves in the lower part that are not attached to the plant

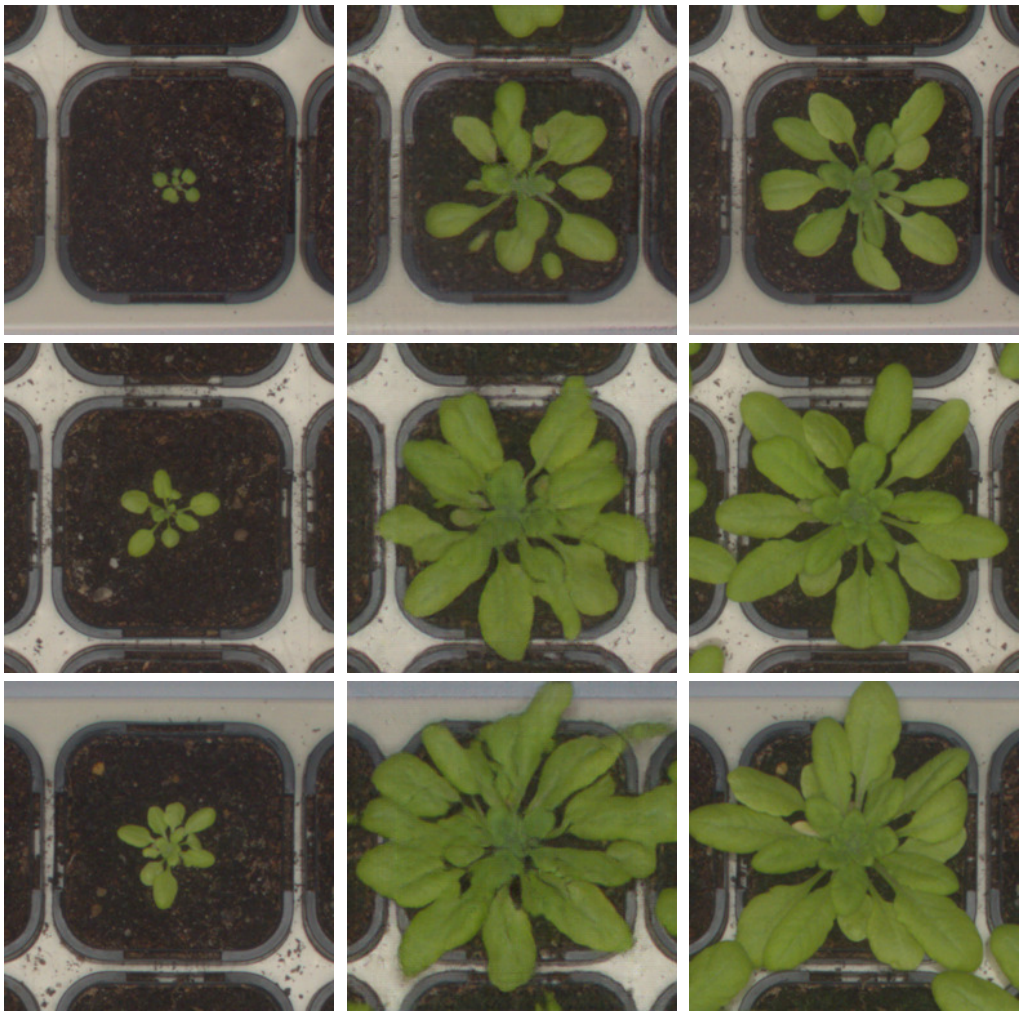
with a petiole, and on day 54 (lower row, center), there is a small artifact in the upper right corner. Likewise, when analyzing the leaf structures, one notices that in all images, fewer petioles have unusual curvatures, and some leaves have uncommon shapes (middle row, center). In terms of color, the generated images are very natural, as the leaves take on slightly different shades of green, which can be all found in the reference, and become slightly lighter from the inside out. Even yellowish leaves or parts of leaves can be found sporadically in the reference as well as in the generated image. In general, it is noticeable that larger outer and inner smaller leaves are generated without being blurry. Another noteworthy aspect is the detailed generation of the background. It looks nearly the same as the input; even the small lumps of dirt change, as indeed it is, their position in the generated image.

Visual analysis of generated Brassica images

For Brassica, the appearance of generated test images of different growth stages is visually assessed with Fig. 5.4. Note that the generated black margin, used in the input to square the images, is cut off since the model was able to generate the area without errors. The generated cauliflower plants of domain B (middle column) look realistic and could be mistaken for real cauliflowers by an unbiased judgment. The comparison of the generated images with reference images for every week (right column) shows a variety of reasons for this. Although there is some noise in some locations, like on the left side of the cauliflower in the generated image of week 9, the overall sharpness of the generated images is almost as good as with the reference. In addition, brightness, contrast, and saturation, as well as color values of foreground and background, match the reference.

A detailed look at the foreground shows that the size, number, and shape of the leaves are plausible. Apart from a few exceptions (bottom leaf in week 5, rightmost leaf in week 8), the orientation of the leaves towards the center of the plant is correct, which can be seen from the direction of the leaf veins. Likewise, the image background is realistically represented. Exceptional brightness levels such as in week 8, in which the background is much darker than in the other images, are captured as well as small details in the background. For example, in steps $1 \rightarrow 4$ WAP and $2 \rightarrow 5$ WAP, the drainage pipes are visible in the generated image at reasonable positions. In the same way, weeds of various sizes are visible in the background next to the cauliflowers in all stages of growth.

When analyzing the relations between the generated images (middle column) and the reference image of input domain A (left column), which is used as a condition for the prediction, we observe a clear correlation for overall cauliflower size. A specific input size in domain A causes a certain output size in domain B, which matches the non-linear growth of cauliflower in the left column. From



domain A: test-input | domain B: generated | domain B: test-reference

Figure 5.3: Example results of experiments with Arabidopsis-P. The left column represents the test-input in domain A, the middle column is the generated output in domain B, and the right column shows the corresponding aligned test-reference image in domain B. A prediction of early growth stages (top row: 23 → 40 DAS), middle stages (middle row: 30 → 47 DAS), and advanced growth stages (bottom row: 37 → 54 DAS) is shown.

weeks 1 to 3, cauliflower shows rather slow growth, which picks up substantially by week 5, when plants show a fast increase in leaf growth and number. Finally, growth slows down again, so there is only a slight size difference between weeks 8 and 9. This rather sigmoid-shaped growth pattern provides a challenge for prediction, as the relation between condition and expected output is not constant over time. However, the resulting sizes in generated images of domain B (middle column) still fit the reference (right column) well.

Looking at the orientation of the individual leaves, there is no obvious pattern between reference images of domains A and B. Although the growth direction and orientation of individual leaves do not drastically change within three weeks, emerging leaves often become more dominant and overlay other plant organs.

Therefore, our temporal growth prediction considers the plant's development as a whole rather than changes in individual leaves. To make use of the generated images, it is essential that the center of plants does not change and is well geo-referenced, just as is expected for plants as sessile organisms. Our example images show that despite inaccuracies of the image alignment (see Sec. 4.2.2), the center position of the plants in the left and middle columns match very well between both domains.

5.3.4 Results of evaluation by instance segmentation

Evaluation of instance segmentation in Arabidopsis-P

From the instance segmentation, the PLA is first derived as the pixel sum of the plants' segmentation mask. A comparison is made between the generated and the reference images by means of the projected leaf area, where Fig. 5.5 focuses on single plants and Fig. 5.6 addresses the daily-average values.

Fig. 5.5 shows a high correlation between the size of the projected leaf area of corresponding generated (y-axis) and reference (x-axis) instances of the same domain. The color indicates sequentially the time of prediction, from dark blue (early growth stage) to green (medium growth stage) to yellow (late growth stage). The high R^2 value of 0.95 indicates a good performance of the model for temporal prediction of plant sizes. Only a slight average overestimation of the projected leaf area can be seen over the whole period, which becomes smaller with increasing plant sizes (regression line slope 0.98). The maximum deviation of the points from the optimal line is about 3000 px in early growth stages and up to 6000 px in later stages, which, taking into account the GSD of 0.32 mm, results in an error in the PLA of approx. ±3-6 cm^2 (5-10 % of image size).

The quality of the temporal prediction is underlined by Fig. 5.6 when comparing the daily-averaged PLA of reference and generated plants in the reference period (days 38 to 56). It is almost the same for every day of the temporal prediction. However, the generated curve is very smooth, while the reference curve has small bumps that are typical of a true plant growth curve. The standard deviation increases the larger the plant becomes, indicating a higher variability in plant size at later growth stages. The maximum deviation of the points from the optimal line is about 3000 px in early growth stages and up to 6000 px in later stages, which, taking into account the GSD of 0.32 mm, results in an error in the PLA of approx. ±3-6 cm^2 (5-10 % of image size). It is noteworthy that the PLA shows a non-linear curve, which means that the PLA gain is properly varying for each prediction, depending on the growth stage.

domain A: test-input domain B: generated domain B: test-reference

Figure 5.4: Examples of input, generated, and reference images of Brassica from 1-6 WAP for domain A and 4-9 WAP for domain B. The rows show different growth prediction steps. $1 \rightarrow 4$ WAP in the top row, $2 \rightarrow 5$ WAP in the second row up to $6 \rightarrow 9$ WAP in the bottom row. The left column represents the input in domain A, the middle column is the generated output in domain B, and the right column shows the corresponding test-reference image in domain B.

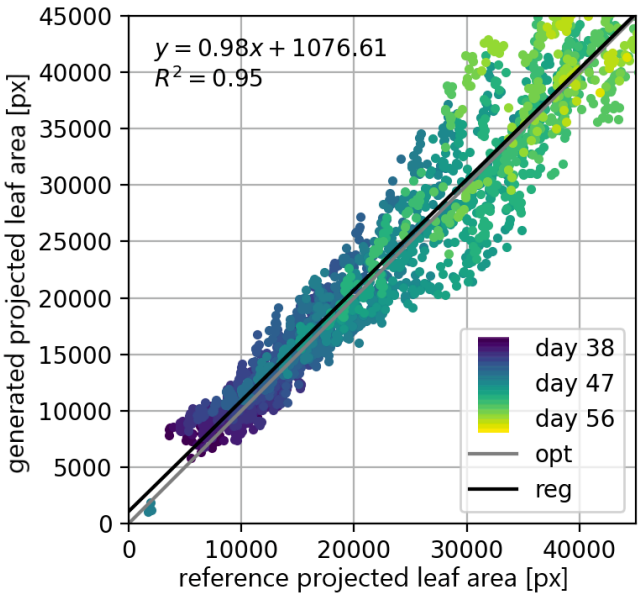


Figure 5.5: Comparison of the projected leaf area [in pixels] of reference and generated *A. thaliana* plants. In the scatter plot, one dot refers to a pair of reference and generated plants. The gray line indicates the optimal line, while the black line represents the regression line. In the upper left corner, the regression-line equation and the R^2 value are indicated.

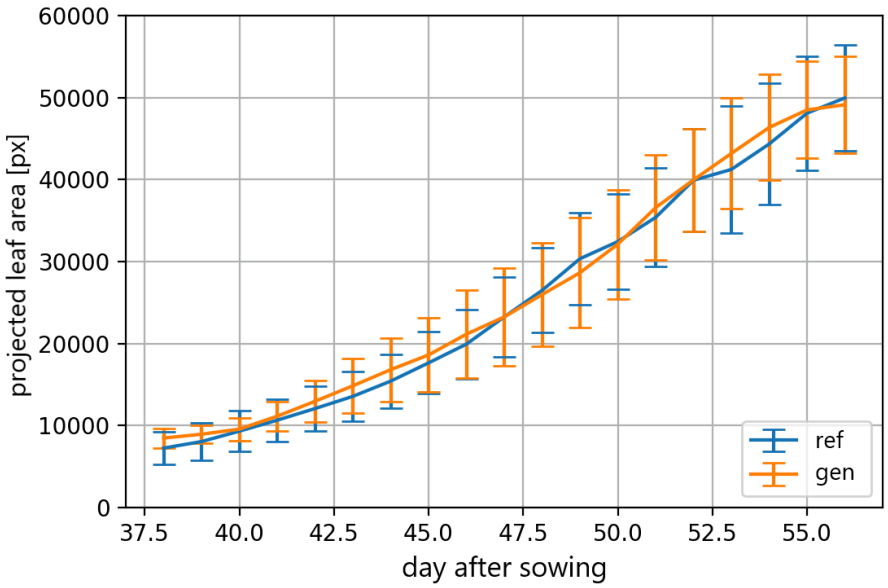


Figure 5.6: Daily-averaged PLA in pixels of generated and reference Arabidopsis-P plants over time. Only the prediction time span (domain B) in the range of 38 DAS to 56 DAS is displayed. The error bars indicate the standard deviation.

Evaluation of instance segmentation in Brassica

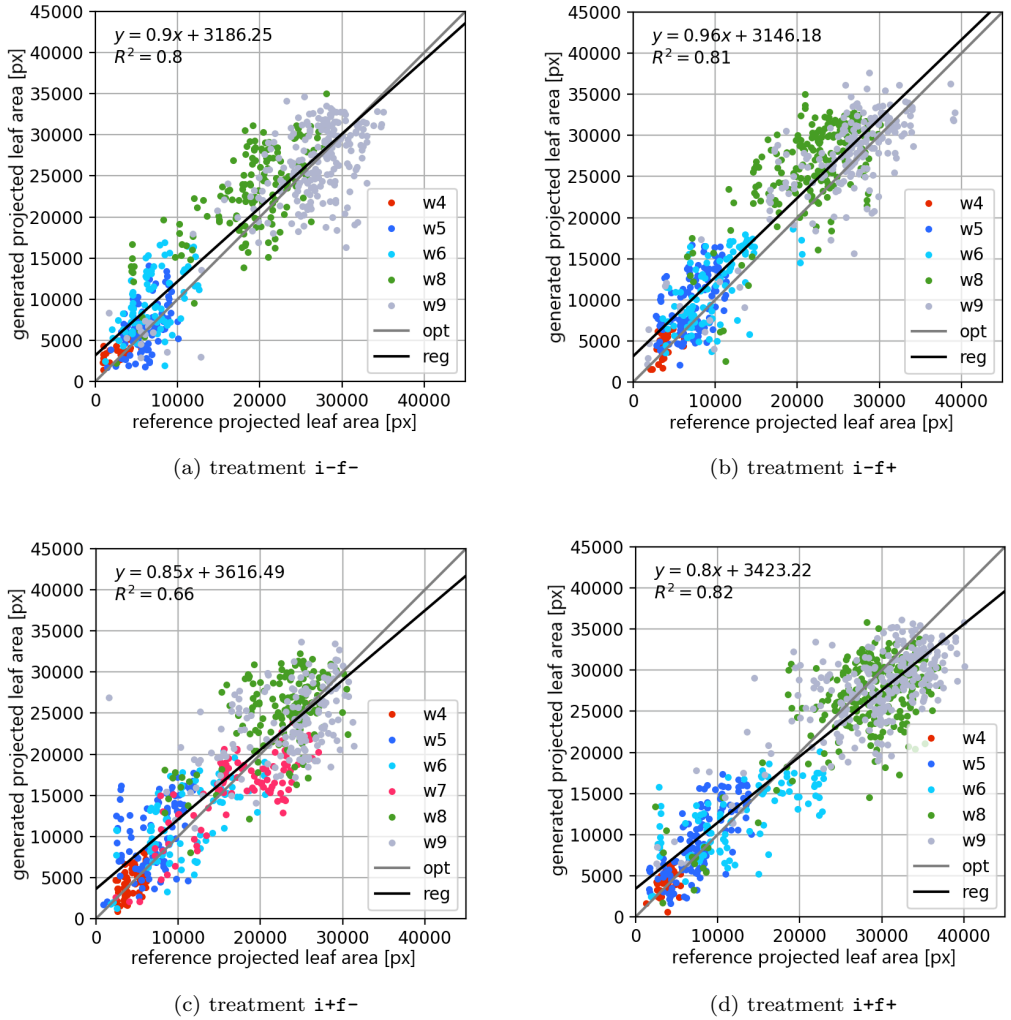
Fig. 5.7 shows the comparison of the growth patterns of generated (y-axis) and reference instances (x-axis) for different treatments. Some findings can be drawn from all four treatments: In all plots, it is visible that the points scatter in an area

around the gray line, which is the optimal line, where the generated projected leaf area is identical to the reference one. Overall, there is a trend that plants in early growth stages are predicted to grow a little too large (black regression line above the optimal gray line), and plants later in growth tend to grow a little too small (black below the gray line). This is also evident from the derived equation of the fitted line, which has a gradient smaller than 1 in all cases. This is likely caused by the observed exponential growth, while the model is trained to work with all plant ages. However, R^2 values from 0.66 to 0.82 show that temporal prediction works well despite the field conditions. It is also noticeable that the points of some weeks are not clearly separated. For instance, week 5 + 6 and week 8 + 9 partially overlap. This is due to the natural variance in the expression of the plants' phenotype; even cauliflowers exposed to the same field treatments develop differently within a certain range. It is also noticeable that the dispersion of the points increases with the plant age, which is explained by the higher natural variance with rising projected leaf area.

Having a closer look, some differences between the treatments can be identified. It is apparent that well-irrigated treatments `i+f-` and `i+f+` are more underestimated than less irrigated treatments `i-f+` and `i-f-`, which can be seen from the number of points below the gray line. Here, it is important to note that irrigation is the most dominant factor influencing plant size [100]. So, large plants are estimated slightly too small, and small plants are slightly too large. We argue that this is due to the joint training with all field treatments, which may cause the different treatments to balance each other out. As a consequence, the model shifts slightly towards the average growth.

Nevertheless, the absolute differences in size between the treatments are well modeled, which is best seen in Fig. 5.8. It shows the averaged PLA for each observation date together with the standard deviation for the reference plants on the left and the produced plants on the right. Only plants located in the image center, as indicated by the center of the bounding box, are taken into account. This avoids plants that are not fully visible in the image and should not influence the distribution of sizes.

It is clearly seen from week 5 onward that the sizes of the reference plants are strongly dependent on the field treatments. Well-irrigated and fertilized plants grow better than plants lacking water and nutrients, which is in line with expectations and analyses of previous studies [100]. The size of reference plants grown under `i+f+` (blue line) was bigger compared to `i+f-` (green), followed by `i-f+` (orange) and `i-f-` (magenta). Although the values of the generated plants with `i+f-` treatment are larger than `i+f+` in weeks 4 to 6, there is a clear analogy to the growth of reference plants under the respective treatments. Noteworthy in week 9 is the bending of the green line in both reference and generated plants.



(a) treatment `i-f-`

(b) treatment `i-f+`

(c) treatment `i+f-`

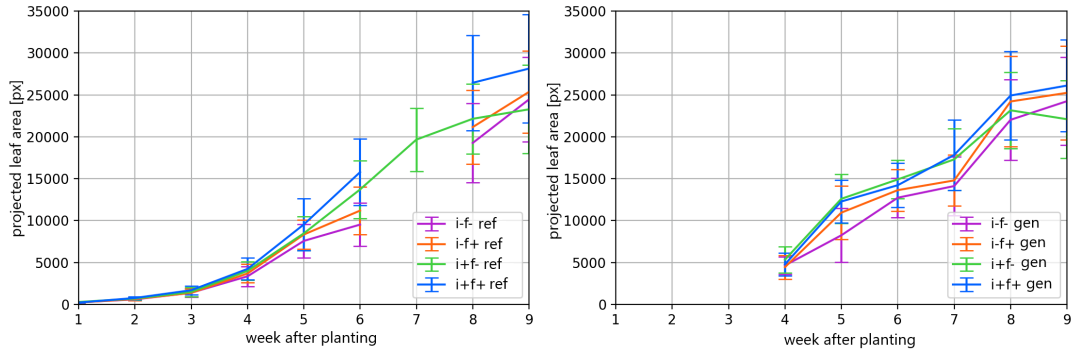
(d) treatment `i+f+`

Figure 5.7: Comparison of the projected leaf area in pixels of reference and generated cauliflower images. The data are separated into four subplots according to their irrigation (`i`) and fertilization (`f`). One dot in the scatter plots refers to a pair of reference and generated plants, where the color of the dots indicates the week (w). The gray line indicates the optimal line, while the black line represents the regression line. In the upper left corner, the straight-line equation and the R^2 value are indicated.

Whatever was inhibiting plant growth at this later stage was apparently already encoded in the week 6 images and was recognized by the model, although the plant sizes were not differently affected at this time (in week 6: line over orange and purple lines).

In week 7, the generated plants show a smaller increase in size in comparison to the other weeks. Using the growth pattern of the reference plants with `i+f-` treatment for comparison, one would expect the generated plants in week 7 to be about 2000 px to 3000 px larger in all field treatments. We see two reasons that it does not occur and that plants are probably often underestimated in week

7. First, the training data for step 4 → 7 WAP (see Tab. 4.1) is missing, and the model incorrectly interprets some input images from week 4 as images from week 3. Second, the beginning of exponential growth at these growth stages causes difficulties, as small differences in the condition have large effects on the generated images. We assume that more training data from the exponential growth period, at best under different climatic conditions, would improve this behavior. However, in all weeks, the generated cauliflower sizes are within the standard deviation of the reference cauliflower sizes.



(a) reference plants

(b) generated plants

Figure 5.8: Development of weekly-averaged PLA [in pixels] between reference (left) and generated plants (right) of Brassica. The error bars indicate the standard deviation in the respective weeks. Curves are separated into different field treatments.

5.3.5 Results of evaluation by Fréchet inception distance

While lower FID is generally better, it is difficult to determine what order of magnitude of FID is expected for a specific dataset and task. It is essential to compare FID values within the context of the experimental expectation, where, in this case, reference plants and generated plants are not expected to match exactly. Therefore, we compare the classical FID between generated and test-reference plants on the one hand with the FID between generated and training-reference plants and, on the other hand, with the FID between the two real distributions test-reference and train-reference. In all cases, the distributions include only the domain of the later growth stage, not the corresponding domain of the earlier growth stage.

For both data sets, $\mathrm{FID}(\mathcal{N}_g, \mathcal{N}_r)$ is in the range from 31.14 to 38.64, which is not an excellent ($\mathrm{FID} < 10$), but a very good image quality. In all cases it is significantly lower than $\mathrm{FID}(\mathcal{N}_g, \mathcal{N}_t)$. This is essential as it suggests that in both experiments, the plants are actually generated from the input conditions rather than replicating the best-fitting training pattern. It can be considered as an indication that the CGMs are capable of generating high-quality images

Table 5.1: Overview about FID scores of both different datasets and treatments. Calculated are FID($\mathcal{N}_g, \mathcal{N}_r$), comparing generated vs. test-reference images, FID($\mathcal{N}_g, \mathcal{N}_t$), comparing generated vs. training-reference images, and FID($\mathcal{N}_r, \mathcal{N}_t$), comparing test-reference vs. training-reference images.

Dataset		FID($\mathcal{N}_g, \mathcal{N}_r$)	FID($\mathcal{N}_g, \mathcal{N}_t$)	FID($\mathcal{N}_r, \mathcal{N}_t$)
Arabidopsis-P		38.12	42.25	28.90
Brassica	`i-f-`	34.18	51.63	55.62
	`i-f+`	33.91	55.29	48.04
	`i+f-`	38.64	56.33	48.21
	`i+f+`	31.14	54.55	49.60

with realistic plant phenotype appearances for both datasets. The FID($\mathcal{N}_r, \mathcal{N}_t$) is lower for Arabidopsis-P and higher for Brassica compared to FID($\mathcal{N}_g, \mathcal{N}_r$). This shows that for Arabidopsis-P, the distributions of test-reference and training-reference have an exceptionally high similarity. We assume that under laboratory conditions, there are fewer variations between the training-reference and test-reference distributions than in the field, where plants are grown under different treatments.

5.4 Discussion

5.4.1 Key factors for realistic image generation

We identified four key factors from the experiments on which the accuracy of the temporal predictions mainly depends, which is essential for applications in the field: The high accuracy of the PLA estimation of Arabidopsis-P shows that, first, a high observation rate and, thus, a large number of aligned images with different time references to training the accgm is beneficial. Second, it shows that exact georeferencing of images is necessary in order to avoid positioning errors in generated images due to incorrect spatial alignment. This is easier to maintain under laboratory conditions, but can also be achieved in the field. Third, the Brassica dataset shows that it is important to get a complete view of whole plants with little overlap of neighboring plants, even at later growth stages. Fourth, high-quality images with sufficient spatial resolution are crucial for detecting nutrient, water, or other deficiencies at an early growth stage. In this way, the model for Brassica was able to detect subtle differences between the treatments in early growth stages that are barely visible to humans (see Fig. 5.8) and was able to generate a correspondingly realistic growth prediction.

5.4.2 Output variability

A common problem when using GANs is mode collapse, which refers to the problem of the model converging to a state where different inputs result in the same or very similar outputs. The number of modes the generator collapses to varies widely, and often, the model jumps back and forth between modes during training iterations. We observed mode collapse in preliminary experiments with both datasets, in which we learned independent models for each time step. For instance, in the Brassica dataset, one model for the time prediction of step $1 \rightarrow 4$ WAP and another model only for step $2 \rightarrow 5$ WAP. Training one model for all growth stages prevented mode collapse, attributed to higher diversity in the training data. However, there are still phenomena that occur similar to mode collapses. In some cases, two generated plants have the same basic structure, but the plant is more extended in a later growth stage. That means the inner leaves look the same, while the outer leaves are expanded. Note that the position of the plant in the generated image remains correct, i.e., even if the inner leaves of the plant appear unchanged, the center position of the generated instance is close to the center position of the input domain. So, the generated plants are still realistic and reasonable, but there is a lack of realistic variability.

In order to reduce these mode collapse similar effects, we have experimented with an increase in input diversity by means of data augmentation methods such as Cutout, CutMix [120], and synthetically generated data. Moreover, we conducted experiments with changed hyperparameters, modifying γ to control the loss weighting (see Eq. 5.3) or choosing a different architecture such as the Diversity-Sensitive CGAN [121], which is designed to force variability through a different structure and loss functionality. None of these attempts were successful in reducing the mode collapse similar effects in the generated images. It shows, that reliably generating images from all parts of the of the given training distributions is a critical issue in generative modeling. However, we present in the next chapters how WGANs are, in certain aspects, capable of tackling this issue.

Another critical issue is the lack of output diversity with variable stochastic components, which is classically a noise vector and in this case dropout. With a change in the stochastic component at test time, the image should also change, but this is hardly the case, as Isola et al. [103] have also found. The extent to which this problem is linked to the mode collapse problem cannot be resolved. With this limitation, it is not possible to visualize variability in the generated images, which would be desirable.

5.4.3 Scalability and limitations

To obtain a comprehensive image-based growth model for cauliflower on a larger scale, it is necessary to increase the amount of data, including more images of plants of different environmental conditions and time points. Images are needed that represent as much variability in input factors and, accordingly, phenotypes as possible so that the diversity in the distribution of images is large but not biased, which can lead to issues with GANs, as discussed in more detail in Sec. 5.4.2. When increasing the set of images, it is important to ensure that plants in all images have similar basic conditions (e.g., type of soil, climatic zone, season, genome). Since there is also a natural bias in plant development in dependence on locations, which is not directly reflected in the images, a new independent model should be trained for each region with deviating basic conditions. In contrast, a model does not need to be limited in time because the distribution of phenotypes becomes particularly diverse over several years due to different environmental influences. If the basic conditions are stable, environmental and management influences can be captured by the CGM from images as the only input, as we demonstrated using different treatments for irrigation and fertilization. However, it could be useful to integrate conditions that are known, such as management decisions, directly into the CGAN. Likewise, the growth stages, which are only implicitly transferred to the CGM in this approach, could also be introduced explicitly. For this, a way must be found to combine temporal and visual conditions. Both ideas are further explored in Chap. 7.

Growth prediction using the presented image-to-image translation method can also be done for plants other than *Arabidopsis thaliana* and *Brassica oleracea var. botrytis* that have structurally different phenotypes, such as wheat or lupines. Similarly, there is no restriction on the number of plants on an image, so plot- or field-wise image generation and growth prediction are also feasible. The more plants there are on each image at a correspondingly lower resolution, the more difficult it becomes to evaluate the generated images by single plant detection, so it becomes necessary to evaluate robust field-wise phenotypic traits, such as the plant number or the total biomass. Another challenge is the height of many crops, where no ground robot with an artificial light source is practical. Alternatives are orthophotos, where, however, lighting conditions are different for each pass. This can be addressed by style transfer methods, also based on GANs [89]. What remains is that with a birds-eye view, only a small part of the plant is observed as the height increases, which is why a change of perspective to a side view could be useful. However, this brings new challenges in the alignment of the images. The finer the leaf structures and grain ears and the lighter the plant, the less it is ensured that the plants themselves do not move between two points in time, and this issue can be amplified by the wind. While geo-referencing and stable

plant positions are crucial requirements for paired image-to-image translation, CycleGAN approaches could help for plants where alignment is not achievable [93], [108].

5.4.4 Implications for agricultural practice

In modern industrial agriculture, generally, a farmer aims to plan already at the planting stage when the field will need to be cultivated and when crops will be ready for harvest. However, uncertain long-term weather forecasts, extreme weather events, and pest or pathogen occurrences make it challenging to predict these outcomes with high accuracy. To account for this, monitoring and screening the plants' current status in the field would be necessary but is labor-intensive and time-consuming if conducted by the farmer or another expert. As presented in this paper, a monitoring approach, which comprises high-throughput sensor measurements and automatic analysis, can overcome several challenges connected to this. First, since the current stage of the plant is continuously observed, and the prediction of the future stage is based on it, the estimated time for harvesting is expected to be more accurate than conventional approaches. Besides, the difference between a plant's current status and a farmer's expectations about plant status is visually accessible and quantifiable, so the farmer can take early action in the field to prevent negative yield results. Finally, planning reliability could be increased, not only for the time of harvest but also for the expected harvest yield.

5.5 Conclusion

In this chapter, we have demonstrated the suitability of a conditional generative adversarial network for temporal crop growth modeling through plant image-to-image translation. By integrating an image of an early growth stage as a condition, we were able to generate an image showing its realistic future appearance. In experiments with laboratory-grown *Arabidopsis thaliana* (Arabidopsis-P) and field-grown *Brassica oleracea var. botrytis* (Brassica), we comprehensively evaluated the generated images two-folded. First, the analysis using Fréchet Inception Distance shows quantitatively that the generated images are of good quality and show strong similarities to the reference images. Second, segmentation masks and plant positions derived from Mask R-CNN show high correlations between projected leaf areas in generated and reference plants. For Brassica, we illustrated that the average plant size and the plant's position are realistically estimated in six different growth stages. Analyzing the results for plants with four different field treatments, we demonstrated that plants subject to good irrigation

and fertilization are predicted to be larger than those with deficiencies, which is consistent with reference measurements. Compared to the laboratory experiment with Arabidopsis-P, we observed a higher discrepancy between generated and reference images, which can be related to the less exact geo-referencing of images and partial overlaps between plants. We consider this method applicable in agriculture because it adds an explainable component to existing CGMs by visualizing the phenotype, is sensitive to tiny differences in crop appearance due to different treatments, and is scalable from single crops to field-wise analyses, as well as from brassica to any other crops. Future image-based CGMs will address existing limitations such as mode collapse-like effects, the less diverse model outputs, and incorporate additional conditions. We further see a high demand for generating a range of possible output images (depending on different input parameters) instead of a single output image, which could support simulations operating within process-based growth models.

Chapter 6

Inter- and extrapolating irregular image time series

When monitoring plants by means of image time series, corrupted, blurred, and missing images are a significant issue as plant growth is no longer holistically analyzable over the growing season [7], [9], [10]. This impacts not only end-users, such as farmers, who plan actions in the field directly based on these images but also researchers and model developers, who may require homogeneous input data and have difficulty dealing with data gaps—especially in machine learning tasks. Sources of such gaps are manifold and, in the case of image acquisition with unmanned aerial vehicles (UAVs) in agriculture, range from an insufficient measurement setup over uncontrollable environmental influences to failures in post-processing calculation of the orthomosaic. In addition, intervals of varying size between observation times are inevitable and even desirable due to non-linear growth patterns, so many observations fall within a range where plants are changing rapidly. Hence, this chapter aims to develop a CGM for the sequential datasets Arabidopsis-S, GrowliFlower, and Mixed-CKA that can replace missing or corrupted images as artificial sensor data from irregular and incomplete sequences (Fig. 6.1).

Compared to the CGMs in the previous chapter, where there was exactly one image in the input, the challenge in this chapter is to handle a non-equidistant sequence of images of arbitrary length. Consequently, a generated image should now be consistent with several images rather than just with one. In addition, we aim for the generation of time-variable output images, i.e., the time of the generating output image should be flexibly selectable, instead of a growth prediction step being predefined for the model.

We introduce the deep generative model TransGrow, a CWGAN that utilizes a CNN for spatial and transformers for temporal modeling. Transformers are attention-based neural network layers suitable for processing and weighting se-

quential input. We demonstrate that with TransGrow, it is possible to explicitly retrieve desired growth stages by including in the model the time points of the input and growth stage to be generated. This differs significantly from comparison methods VAEs and Adversarial Autoencoders (AAEs), which interpolate in latent space for this purpose and thus have only implicit retrieval. One advantage of explicit retrieval is the possibility of test-time extrapolations: So, in addition to interpolations for data imputations, probable past and future expressions of the above-ground phenotypes can be generated.

Apart from time, no other growth influencing factors are taken into account. Nevertheless, we consider plant growth not to be deterministic and focus on generating a realistic image distribution to indicate which plant parts have a high generation variability in growth development. Compared to the previous chapter, where a classical CGAN was used, the change to a CWGAN is particularly intended to ensure that the stochastic model component is not suppressed, as discussed in Sec. 5.4.2. To better understand the model in general and the impact of the stochastic component in particular, we analyze the position of image embeddings in the model's latent space. In addition to plant traits derived from generated Arabidopsis-S images, this leads to a more explainable model.

Summarizing the key contributions of this chapter:

- Flexible and realistic image generation using a CNN-Transformer conditioned by an input image sequence of arbitrary length and time and by the output time, i.e., the requested growth stage of the image to be generated.
- The possibility to perform test-time extrapolations in addition to interpolations for data imputations in order to predict probable future expressions of the above-ground phenotypes.
- The generation of distribution for each point in time and, from this, the visualization of predictive variability on the plants in the generated image.

6.1 State of the art

6.1.1 Different ways of image imputation

Interpolation approaches for data imputation

In order to interpolate between two reference images, traditional and most intuitive approaches directly operate in the image space, such as linear transformations, image warping, or optical flow. However, the resulting images are often

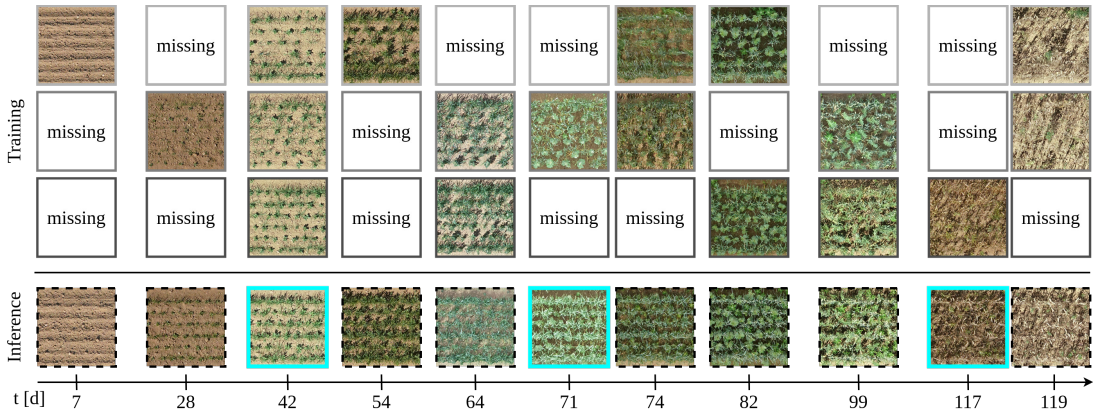


Figure 6.1: The aim is to learn a realistic spatio-temporal model of plants from several irregular and incomplete sequences during training in order to replace missing images (dashed images) from a different number of input images/support points (cyan images) for inference.

inaccurate and not appealing since only the appearance, but no underlying features are changed for generation. This problem is addressed by mapping the essential features of the input images into a latent space and generating new images by interpolation between latent codes, for instance, enabled with AE and their variants with adversarial constraints [122], [123]. Still, there are three major issues: First, time is not explicitly taken into account, so assumptions about the growth process have to be made, e.g., linear interpolation, which does not hold for plant growth in the image space. Second, the additional modeling of predictive uncertainties by extending the model with a stochastic component is missing. This is essential since the observations never capture all complex growth influencing factors. Third, extrapolations remain challenging, which are needed, for example, to derive information from a plant's potential future development, such as the estimation of harvest yield and date.

Conditional generative modeling for data imputation

More sophisticated methods, such as various types of variational autoencoders (VAE) [30] or conditional GANs [12], [124] overcome the aforementioned limitations by forming a distribution in the latent space, which allows controlled sampling. These distributions can be spanned by various combined conditions, such as stochasticity paired with images, categorical labels, or text. However, when there are sequential temporal conditions, as in this paper, most work is based on at least one of the following assumptions: Either time is modeled implicitly and is not controllable so that only a distribution for a specific point in time is generated, for example, the exclusive generation of future plant phenotypes from a fixed set of previous images [14], [97], [125]. Alternatively, the temporal component is inherently accounted for by equidistant or concise input intervals,

such as in video frame prediction, which is nearly impossible for real-world sensor measurements in agriculture [98], [126], [127]. Therefore, we propose temporal positional encoding in combination with a transformer encoder [82] to control the time factor explicitly.

6.1.2 On temporal modeling with transformer

To address the problem of non-equidistant inputs, image timestamps are used for positional encoding as in [128]. We call this global positional encoding since the acquisition times of the images related to a dataset reference point are used as positions. These positions deviate significantly from classical absolute positions, where the transformer is provided with indices related to their order in the sentence [82], positions of image patches related to their location in the original image [129], or with the relative timestamp of video frames [127]. Thereby, for works from the video domain, the spatial and temporal encoding is either directly combined [130], [131] or initially decoupled [132]. We follow the latter idea to encode both independently and to merge them just before the transformer layer, thus creating the weighting of spatio-temporal image embeddings by self-attention.

6.2 Methods

In this section, we introduce the framework, which we call TransGrow[1], before giving more details about the linking of CNN and transformer in the generator and introducing evaluation and comparison methods.

6.2.1 Framework for image generation in time series

The TransGrow framework is a CWGAN whose generator is characterized by a combined CNN-Transformer architecture to obtain spatio-temporal image embeddings of irregular sequences of different lengths and to generate realistic images from them. While the different architectural components of generator $\mathcal{G}_\theta$ and discriminator $\mathcal{D}_\delta$ have been adopted from various state-of-the-art models, the explicit integration of time via positional and transformer encoder, allowing the generation of images of arbitrary time points is a novelty in the GAN context. Compared to the CWGAN in Chap. 7, the structure differs considerably, and further regularizations have been added to the objective to accelerate convergence.

[1]Source code is publicly available at `https://github.com/luked12/transgrow`

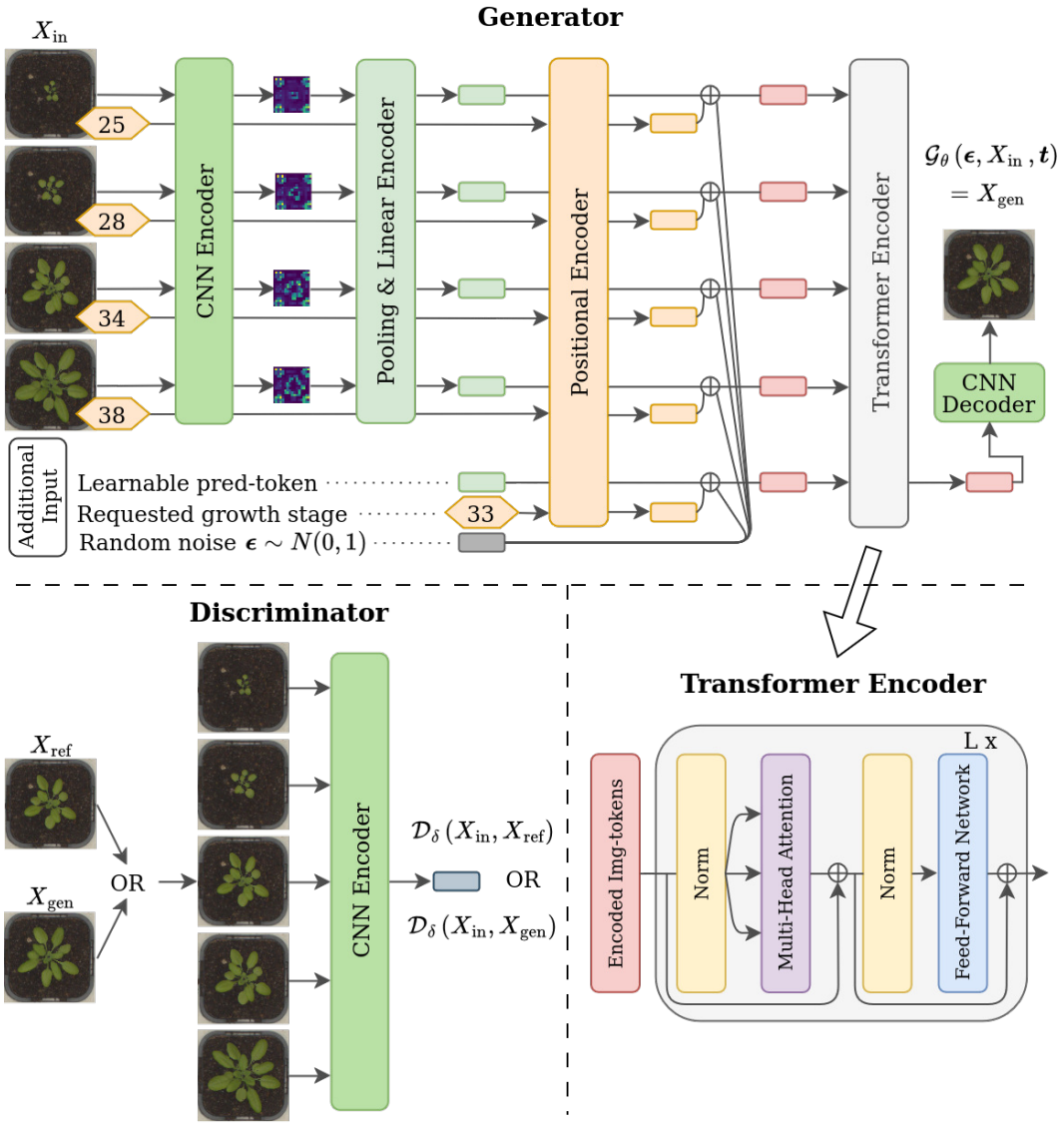


Figure 6.2: TransGrow framework. In the generator, an input sequence X_{in} is first spatially encoded with CNN, pooling and linear layer to dimension d_{model} and then, along with times $\boldsymbol{t}$ and stochasticity $\boldsymbol{z}$, temporal encoded within positional encoding and a transformer stack. Out of this, an image of the requested growth stage is encoded. The CNN-based discriminator takes the input sequence with either the reference X_{ref} or the generated image X_{gen} sorted into the correct chronological position.

Generator

In the generator, a target image $X_{\text{gen}} = \mathcal{G}_\theta(\boldsymbol{\epsilon}, X_{\text{in}}, \boldsymbol{t})$ is generated from noise $\boldsymbol{\epsilon}$, an input image sequence X_{in} consisting of I_{in} images, and times $\boldsymbol{t}$, whereby $\boldsymbol{t}$ is in fact split into $[\boldsymbol{t}_{\text{in}}, t_{\text{gen}}]$. According to Eq. 2.12, the CWGAN conditions y are made up of X_{in} and $\boldsymbol{t}$. To get X_{in}, an image sequence sample X of length $I = I_{\text{in}} + 1$ needs to be divided into the input image sequence $X_{\text{in}} = [X_1, \ldots, X_{I_{\text{in}}}]$ with associated times $\boldsymbol{t}_{\text{in}} = [t_1, \ldots, t_{I_{\text{in}}}]$ and an image to be generated X_{ref} with its requested time t_{gen}. For this purpose, the position of the image to be generated is

randomly sampled at each training step. To indicate the time differences between the input images and the target image, we introduce the vector $\Delta \boldsymbol{t} = |\boldsymbol{t}_{\text{in}} - t_{\text{gen}}|$, where it is expected that the temporal closest input image with distance $\min \Delta \boldsymbol{t}$ has the highest influence on the generation of $\boldsymbol{X}_{\text{gen}}$.

Discriminator

In the discriminator, either the reference $\mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \boldsymbol{X}_{\text{ref}})$ or the generated image $\mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \boldsymbol{X}_{\text{gen}})$ are presented along with the input image sequence. Before feeding through the discriminator, the target image is first sorted into the correct position of the input sequence. So, the discriminator does not require time point embeddings; rather, it considers the overall consistency in the growth development. It is built as a lightweight model from alternating convolutional layers, ReLU activation, instance normalization, and weight-sharing between all images in the sequence. The output is a score specifying how realistic the input is, suitable for enforcing the minimization of the Wasserstein distance within the following adversarial objective.

Conditional Wasserstein GAN objective

The adversarial objective is to optimize the parameters θ and δ by maximizing several combined objective functions L by $\mathcal{D}_\delta$ and minimizing them by $\mathcal{G}_\theta$.

$$\begin{aligned} \theta^*, \delta^* = \arg\min_\theta \arg\max_\delta & \mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta) \\ & + \lambda_{\text{L1}} \mathcal{L}_{\text{L1}}(\mathcal{G}_\theta) + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}}(\mathcal{G}_\theta) + \lambda_{\text{VGG}} \mathcal{L}_{\text{VGG}}(\mathcal{G}_\theta) \end{aligned} \tag{6.1}$$

Here, $\mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ represents the total CWGAN-GP objective function [35] while the other terms are further regularizations applied to the generator. Those are added with corresponding weights λ to the final objective in the form of a $\mathcal{L}_{\text{L1}}$ reconstruction loss, a multiscale structural similarity (SSIM) loss $\mathcal{L}_{\text{SSIM}}$ [39], and a perceptual content loss employing a pretrained VGG-network $\mathcal{L}_{\text{VGG}}$ [133].

Eq. 6.2 represents $\mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ the classic CWGAN objective [34], added with the gradient penalty [35], as in Eq. 7.2.

$$\begin{aligned} \mathcal{L}_{\text{CWGAN}}(\theta, \delta; \boldsymbol{X}_{\text{in}}, \boldsymbol{X}_{\text{ref}}, \boldsymbol{\epsilon}, \boldsymbol{t}) = & \mathbb{E}_{(\boldsymbol{X}_{\text{in}}, \boldsymbol{\epsilon}, \boldsymbol{t})}[\mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \mathcal{G}_\theta(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{t})] \\ & - \mathbb{E}_{(\boldsymbol{X}_{\text{in}}, \boldsymbol{X}_{\text{ref}})}[\mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \boldsymbol{X}_{\text{ref}})] \\ & + \lambda_{\text{GP}} \mathbb{E}_{(\boldsymbol{X}_{\text{in}}, \hat{\boldsymbol{X}})}[(\|\nabla_{\hat{\boldsymbol{X}}} \mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \hat{\boldsymbol{X}})\|_2 - 1)^2] \end{aligned} \tag{6.2}$$

For the calculation of gradient penalty, $\hat{\boldsymbol{X}} = \eta \boldsymbol{X}_{\text{ref}} + (1 - \eta) \mathcal{G}_\theta(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{t})$ represents a randomly weighted average between the generated and the reference image, so η is picked randomly between 0 and 1. The weighting of the whole term is done by coefficient λ_{GP}. It has been shown that $\mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ is able to minimize the Wasserstein-1 distance between the distribution of the real and generated samples.

Optimization tricks

Since the sequence length in the datasets varies from plant to plant, no uniform sample sizes necessary for efficient model training in batches are given. In addition, the training is computationally inefficient the longer the input sequences because the data accessing time increases more compared to the forward pass time. To overcome these issues, the number I of randomly sampled images forming an image sequence X is significantly lower than the number of images ${}^{k}J$ of an entire sequence of a plant k. While $\varnothing J$ varies between 8.50 and 849.75 for the sequential data sets (see Tab. 4.3), we set $I = 4$ for the training of all datasets. Subsequently, we randomly choose the target image out of the sequence, aiming to generate interpolations if $\min(\boldsymbol{t}) < t_{\text{gen}} < \max(\boldsymbol{t})$, and extrapolations if $t_{\text{gen}} = \min(\boldsymbol{t})$ or $t_{\text{gen}} = \max(\boldsymbol{t})$, in random alternation. This is crucial because it contributes to the positional encoding not being suppressed during the training but having an actual impact on the time of the generated image. Without random sampling, TransGrow would only be able to generate the position of a sequence previously defined in training, which is often the case for video prediction, where the target image is always the last image of the sequence. While training is performed with a fixed sequence length, at inference, the sequence length and the target image can still be varied as required.

6.2.2 Combining CNN and transformer

The following paragraphs provide details on how CNN and transformer are linked in the CWGAN generator for spatio-temporal image encoding and decoding.

Spatial encoding using CNN

For the spatial encoding of X_{in} a ResNet-18 backbone [42] is used as CNN encoder, which is pre-trained on ImageNet [134]. It represents a comparatively small and thus efficient trainable model (see Sec. 2.5.2) whose embedding dimension (512) after pooling the last convolutional layer also represents a reasonable latent dimension for the transformer encoder. We encode each image independently with the same ResNet-18 weights so that independent features result from the same latent space. This means the sequence length can be flexibly varied without affecting the number of model parameters. In order to preserve experimentation flexibility for different latent dimensions, the pooled CNN output is linearly projected to a final latent dimension d_{model}.

Weighting image embeddings using transformer

The architecture's transformer module is a stack of multiple transformer encoder layers [82], ensuring temporal connections of the sequence elements by increasingly deeper self-attention. Each layer contains a Multi-Head Self-Attention (MSA) and a Feed-Forward Network (FFN), with layer normalization before each one and skip connections around each one (bottom right in Fig. 6.2). Like the CNN encoder, the transformer encoder can accept any number of sequence elements, which is crucial for TransGrow's flexibility in processing different numbers of input images. It generates an output, called memory, of the same dimension d_{model} for each incoming vector of the sequence, of which only the transformed pred-token is further used.

Positional encoding

In the positional encoding, the spatially encoded latent representations of the input images are provided with explicit temporal information by adding I_{in} positional encoding vectors of the same dimension d_{model}. Besides, the requested growth stage in the form of a positionally encoded time point is added to the learnable pred-token. We use a global positional encoding, whereby one dataset-specific reference point is set to a reasonable time at the beginning of the growing period, such as the date of sowing or planting. Thus, the global position of each image can be interpreted as the plant's growth stage. This enables the transformer to capture the actual temporal information of the sample in the dataset, which is required to generate images at arbitrary points in time instead of generating fixed positions with respect to the input sequence. The positional encoding is calculated by a combination of sine-cosine signals, as in [82].

$$\boldsymbol{p}_{(t,2i)} = \sin\left(\frac{t}{10\,000^{2i/d_{\text{model}}}}\right) \quad \boldsymbol{p}_{(t,2i+1)} = \cos\left(\frac{t}{10\,000^{2i/d_{\text{model}}}}\right) \tag{6.3}$$

Here, t represents the global temporal position (i.e., the growth stage), and i is the positional dimension.

Learnable pred-token

Inspired by [129], we add a d_{model}-dimensional learnable token, called pred-token, from which the target image is decoded. Besides, it is intended first, to carry basic shared features of all samples of the dataset and second, to have a container to which the positional encoded requested time t_{gen} is added and brought to the transformer encoder.

Inducing stochasticity

To be able to generate an output distribution at fixed $\boldsymbol{X}_{\text{in}}$ and $\boldsymbol{t}$, stochasticity in the form of random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$ is induced into the network. Therefore, the identical $\boldsymbol{\epsilon}$ of dimension d_{z} is scaled up to d_{model} and then added to all latent representations of the sequence together with the positional encoding. While there are various other methods to induce noise, this one has proven to be very robust in our case. To achieve force diversity, additional dropout layers are used after the positional encoding and within the transformer encoder at training time.

Generator decoder

As a generator, we use a lightweight decoder [135], which, in particular, is designed to be robustly trainable with small datasets. It includes an initial ConvTranspose layer followed by BatchNorm and GLU to decode from $1^2 \rightarrow 4^2$. The subsequent upsampling modules, each quadratically enlarging the image, consist of Nearest-Upsampling, Conv, Batchnorm, and GLU layers. To counteract the weak gradient flow with deep decoders, an additional skip layer excitation between the images $8^2 \rightarrow 128^2$ and $16^2 \rightarrow 256^2$ is used.

6.2.3 Evaluation of generated images

We utilize the established metrics MS-SSIM and PSNR to compare generated and reference images of the same time point in the image domain and FID to compare the generated and real image distribution across all time points in the feature domain, as introduced in Sec. 2.5.1. While for SSIM (opt: 1) and PSNR (opt: $\sim$60 dB), the larger, the better; for FID, a smaller value means a higher similarity between the image distributions (opt: 0).

For Arabidopsis-S, we additionally perform a plant-trait-based evaluation by determining the PLA. We use the vegetation index RGBVI [83] and calculate a leaf area mask to which every pixel with an RGBVI>0.25 is added. The Arabidopsis-S dataset is well suited for the application of the RGBVI because the Arabidopsis-S plants represent the only green pixels in front of a homogeneous, non-vegetation-like background. In addition, a short calculation time is advantageous because the masks on the real training images are also used for flexible data augmentation of the Arabidopsis-S dataset. Therefore, a fast and efficient vegetation index is preferred over deriving the PLA from MaskR-CNN segmentations, which has a higher inference time, including the filtering of suitable masks.

6.2.4 Comparison methods

While with TransGrow, the requested growth stage is specified in the input, and target images can thus be generated explicitly, it is also possible to interpolate between the input images. This chapter considers both linear interpolations in the image space (L1) and the latent space using VAE [30] and AAE [136]. For linear interpolation, we calculate the difference matrix in the image space or the difference vector between image embeddings of two support points in the latent space. We consider the length of this difference vector as the total time difference and interpolate this according to the desired target time. This method is intuitive, widely used for sampling from latent spaces in different generative models like GANs and AEs, and remarkable can be obtained [123], [137]. It has been demonstrated to create smooth transitions between representations of different domains, classes, or poses of a dataset [138], [139]. Furthermore, they are suitable for generating a variance in the output that follows a predefined distribution - here $\mathcal{N}(0, 1)$. However, unlike TransGrow, they do not allow time-dependent sampling of the latent space and, therefore, no intuitive extrapolation, as further discussed in Sec. 6.4.1.

For maximal comparability, VAE and AAE are provided with the same generator as TransGrow. In the encoder, only the transformer stack is omitted because, for training VAE and AAE, the data are not treated as sequential datasets $\mathcal{S}$ but as classic image datasets $\mathcal{X}$ (see Sec. 2.2). So, no time component is involved, and thus, no explicit temporal encoding is required. Instead of the transformer stack, the VAE-typical d_{model}-dimensional bottleneck with μ and σ is synthesized by a linear layer and decoded to the image after reparametrization from the normal distribution. In AAE, a discriminator is used with three linear layers, each followed by ReLU and final sigmoid activation. In addition to an L1-loss for VAE, the KL-divergence and for AAE, an adversarial loss utilizing binary cross entropy are used for optimization. Both models are trained until the l1 losses converge on the validation images.

6.3 Experiments and results

6.3.1 Experimental setup

Data augmentation and preparation

Since transformer training benefits from high data diversity, multiple augmentations are performed on the train sets consisting of random 90°-rotations and horizontal and vertical flipping. After augmentation, images of all datasets are scaled to the value range $[0, 1]$.

In addition, and especially for the Arabidopsis-S dataset, comprehensive foreground (plant) and background (pot) "shuffling augmentation" is implemented, which significantly increases the diversity of the dataset. To achieve this, the Arabidopsis-S plants are first cut out using the RGBVI [83], then rotated arbitrarily, and finally placed randomly on one of a set of eight selected background images of an empty pot. Note that for the rotation, the rotation axis is the image center and not the plant center, which does not match exactly, so the position of the plant center on the augmented image varies slightly. This special shuffling augmentation is applied to training, validation, and test images. Although this limits the generalizability and the comparability of the Arabidopsis-S experiments with experiments without shuffling, augmenting only training and validation data leads to considerable difficulties for two reasons: First, the shuffling establishes a constant background over time, which is not the case in the non-augmented test set due to moving particles and lumps of dirt in and next to the pot. Therefore, a distribution shift would be introduced between the training and test sets. Second, since the number of available images of empty pots is severely limited, shuffling augmentation overfits the eight given background pots and thus also generates them at test time. However, comparing the original backgrounds with the generated ones from the shuffling augmentation would have undesirable effects on non-plant-trait-based evaluation metrics.

Model hyperparameter

The stochasticity and the embedding dimension are set to $d_\mathrm{z} = 16$ and $d_\mathrm{model} = 512$, as the latter is the size of the last ResNet-18 feature layer. A low dropout probability of 0.1, as suggested in [82], [129], is used at all dropout positions as described in Sec. 5.2 to prevent overfitting. Since the transformer is intended to encode only the temporal component of the input, a low depth $L = 3$ and a number of four heads within the multi-head attention are experimentally selected. In the calculation of the generator loss, the additional regularizations are all equally weighted $\lambda_\mathrm{L1} = \lambda_\mathrm{VGG} = \lambda_\mathrm{SSIM} = 1$. While in classical GANs, the reconstruction loss (here: L_L1) is often weighted substantially higher than L_GAN [14], [103], this did not turn out to be beneficial in the experiments. The weighting of gradient penalty $\lambda_\mathrm{GP} = 10$ and all other CWGAN-GP optimization hyperparameters are set according to [35]. Using a batch size of 32, a learning rate of 1e-4, Adam optimizer, and the settings mentioned above, it takes up to 1000 epochs for convergence, running for approximately 5 d on a single Nvidia A100 in mixed-precision mode.

Table 6.1: Comparison of TransGrow with classical interpolation methods in image space (L1) and latent space (VAE, AAE) using the metrics MS-SSIM, PSNR [db], MAE_{PLA} [% image^{-1}], and FID.

Approach	MS-SSIM (↑)	PSNR (↑)	MAE_{PLA} (↓)	FID (↓)
L1	0.89 ± 0.09	**31.64** ± 4.92	2.02 ± 2.42	**11.35**
VAE	0.87 ± 0.07	28.91 ± 2.54	1.00 ± 1.06	54.79
AAE	0.88 ± 0.09	28.83 ± 3.06	1.83 ± 2.74	20.11
TransGrow	**0.90** ± 0.05	29.15 ± 2.28	**0.69** ± 0.61	13.92

6.3.2 Comparing TransGrow with linear interpolation methods

In the first experiment, we investigate the interpolation capabilities of TransGrow and compare them with the baseline methods L1 (image space), VAE, and AAE. For this comparison, we use the Arabidopsis-S dataset, sample three random images from the same time series, and interpolate the temporal middle image as the target image from the other two. In total, 200 times $I = 3$ random images are sampled from each of the 16 test time series, resulting in a total of 3200 target images from which scores in Tab. 6.1 are calculated. So, there is a random interpolation distance for each generation. The difference between the latest and earliest growth stage 38 DAS - 21 DAS = 17 DAS means that the nearest input image is a maximum of 8 d away, i.e. $\min \boldsymbol{\Delta t} = 8$. The model time unit of TransGrow is set to whole days (discussion on this in Sec. 6.4.2), which is why the interpolation in the baselines is in day increments as well. Since a sequence includes several images from the same day, the reference image to be interpolated can show small deviations from the input image, although they have the same time stamp. These deviations can occur in the order of magnitude of an Arabidopsis-S growth development of a maximum of one day and affect all methods equally.

Quantitative results

In Tab. 6.1, the interpolation capabilities of the methods are compared using the metrics MS-SSIM, PSNR, MAE_{PLA} and FID. For the MS-SSIM, all values are in the range between 0.87 and 0.9, with TransGrow performing best, followed by L1, AAE, and VAE. In terms of PSNR, L1 performs best with 31.64 db, followed by TransGrow, VAE, and AAE worst with 28.83 db. The high standard deviations for both MS-SSIM and PSNR indicate large differences between the interpolation distances, which is why this is examined in more detail below. In the PLA, TransGrow clearly has the lowest deviation with $\text{MAE}_{\text{PLA}} = 0.69$, followed

by VAE (1.00), AAE (1.83) and L1 (2.02) and sets itself apart significantly from AAE and L1. With the FID score, there are large differences; it is high for VAE (54.79), significantly lower for AAE (20.11), and at a similarly low level for TransGrow (13.92) and L1 (11.35). Overall, L1 and TransGrow have the best and most similar values, and an MS-SSIM of ~ 0.9, a PSNR $\sim 30\,\mathrm{db}$, and an FID<15 indicate a high image quality. However, the clear discrepancy with $\mathrm{MAE}_{\mathrm{PLA}}$ is apparent, i.e., the PLA calculated from L1 interpolated images is significantly less accurate than with TransGrow. This is due to blending effects in the image space, in which not only the interpolated plant but also the plants from the two input images shine through, as can be seen in the following qualitative results.

Qualitative results and visualized variability

A qualitative comparison between the methods is given in Fig. 6.3 for the Arabidopsis-S dataset. The interpolations for L1, VAE, and AAE are performed from the two nearest input images, and the images for TransGrow are calculated from $I_{\mathrm{in}} = 3$ input images. In addition, a variability image is displayed for each generated time point by plotting the pixel-wise standard deviation from generations with ten different random stochastic components $\boldsymbol{\epsilon}$ and otherwise constant input. In the top line, L1 image space interpolation is visualized. Here, it becomes apparent that blending between leaves of different growth stages does not work well since leaf contours of both early and later growth stages are visible, and not the actual leaf structures are formed. This is the main reason for the worse $\mathrm{MAE}_{\mathrm{PLA}}$ score (Tab. 6.1) and makes the method inappropriate for any phenotyping applications. In addition, the interpolation is deterministic, so that no variability image can be generated.

The plants generated by AAE, VAE, and TransGrow very well match the reference images in terms of plant size, number of leaves, and leaf orientation. A continuous growth behavior over time can be found in all approaches. However, there are also differences: While the VAE plants are a bit blurry, which is a typical problem of VAE [140], the AAE plants have sharp edges, but the leaves appear a bit too clunky. There is a lack of texture on the leaves, and in some places, leaf petioles are missing (e.g., 28 and 31 DAS). In comparison, the TransGrow images have more texture and the best overall appearance. Artifacts occasionally occur in all methods, especially with rapidly developing leaves or when the interpolation distance is high, such as for the images at 33 DAS. This refers to unnaturally shaped leaves or leaves that have no connection to the plant center.

The variability maps also reveal significant differences, with a darker blue indicating a larger pixel-wise standard deviation. First of all, it is noticeable that in all methods, the variability occurs exclusively on the plants themselves and not on the background, which is due to the shuffling augmentation. Focusing on

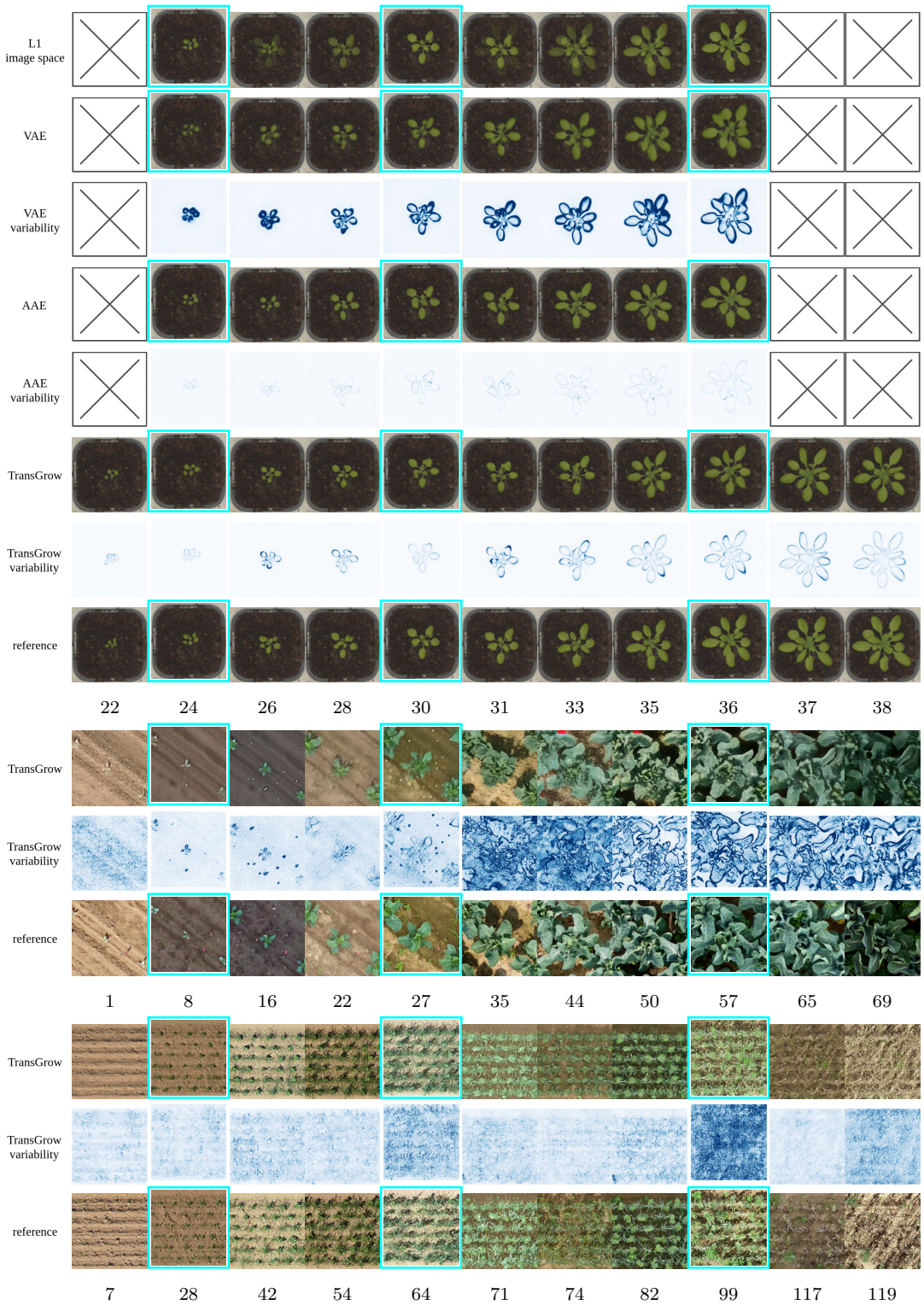
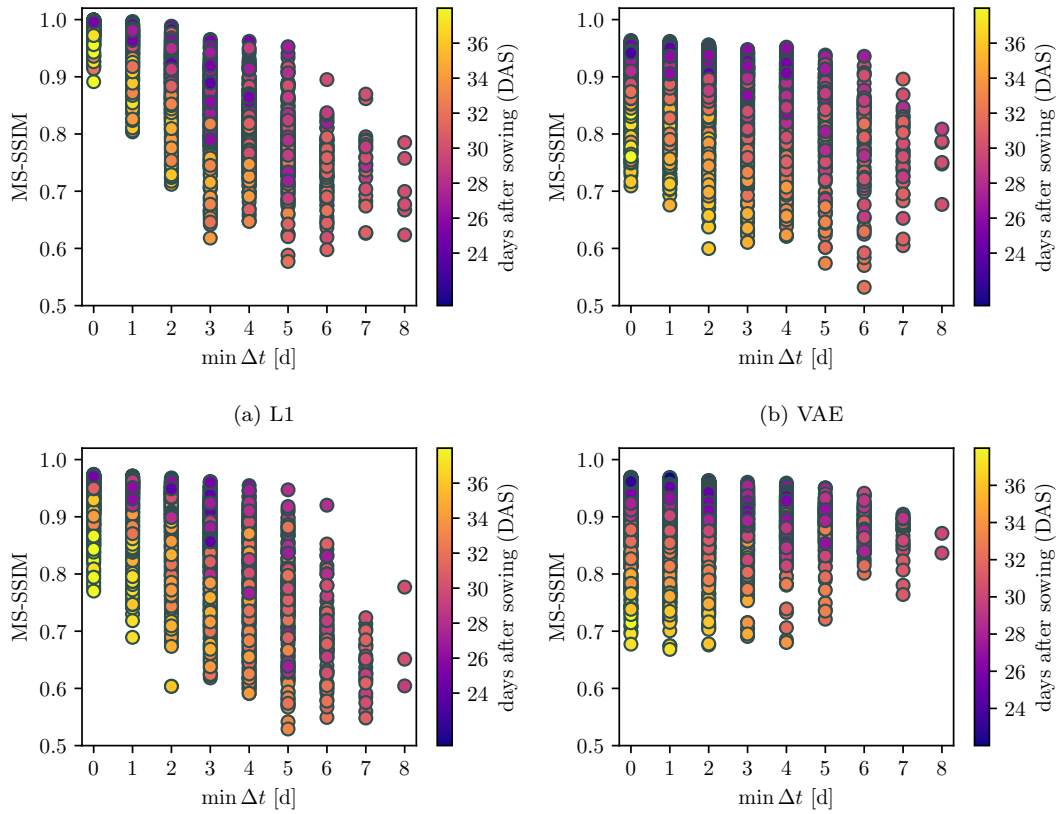


Figure 6.3: Generated images and associated variability maps over time. On the top is the comparison for Arabidopsis-S between L1 image space interpolation, VAE, AAE, and TransGrow. In the middle and below are the TransGrow-generated images for GrowliFlower and Mixed-CKA. While baseline methods allow only interpolation, with TransGrow and $I_{\text{in}} = 3$ input images (cyan frame), extrapolation in the past and in the future are shown.



(a) L1 (b) VAE

(c) AAE (d) TransGrow

Figure 6.4: MS-SSIM of generated Arabidopsis-S images in relation to temporal nearest support point (min Δt) and growth stage indicated by the color of dots for the different methods (a) L1, (b) VAE, (c) AAE, and (d) TransGrow.

the plants, it is realistic that in all methods, the variability is greatest at the leaf edges. VAE exhibits significantly greater variability than TransGrow, which in turn exhibits greater variability than AAE. Remarkably, VAE and AAE have an almost equally distributed variability on the leaf surfaces, although not all leaves develop at the same rate. On the contrary, TransGrow's leaf edge variability is more nuanced, and the images are particularly darker blue if they belong to fast-growing leaves. This is explainable as the difference between the respective closest support points is particularly large in these cases.

Impact of the interpolation distance

While Tab. 6.1 shows evaluation metrics averaged over a set of random interpolation distances, the standard deviations indicate that there is a strong difference in image quality depending on the interpolation distance. Since the image to be interpolated does not necessarily lie exactly in the temporal middle between two input images, we analyze the MS-SSIM in Fig. 6.4 depending on the temporally closer input image min Δt. In addition, the growth status of the image to be in-

terpolated is visualized in color according to the colorbar on the plot's right-hand side. Two aspects all approaches have in common: First, at the same growth stage, MS-SSIM decreases with increasing $\min \Delta \boldsymbol{t}$, i.e., the greater the interpolation distance, the poorer the image quality. Although the quality of L1 drops significantly faster than the other methods. Second, it can be noted that younger plants have mostly a higher MS-SSIM than older plants. This is due to the fact that the easy-to-generate background portion is larger for younger plants.

Overall, all images generated for TransGrow are at a higher level with MS-SSIM > 0.65 than VAE and AAE with MS-SSIM > 0.55. However, TransGrow has the worst scores at $\min \Delta \boldsymbol{t} < 2$, especially for plants at later growth stages. In particular, identity mapping has values MS-SSIM < 0.7 and is thus worse than L1, VAE, or AAE. If there was only one image for each day, a significantly higher identity mapping score could generally be expected for all approaches and MS-SSIM $= 1$ for L1. Remarkably, for particularly long interpolations of $\min \Delta \boldsymbol{t} < 5$, the TransGrow's MS-SSIM show significantly better results than the baseline approaches (up to an MS-SSIM difference of 0.3). Naturally, interpolation in latent space carries a greater risk of leaving the data manifold as the distance between the support points increases, whereas this risk is lower with TransGrow due to explicit access of the latent space, as visualized by image embeddings in latent space (Sec. 6.3.4).

6.3.3 Inter- and extrapolation across different datasets

Besides interpolation, it is also possible to extrapolate with TransGrow. In this section, three aspects will be examined in more detail. First, the ability to perform interpolation and extrapolation across different sequential datasets GrowliFlower and Mixed-CKA and the potential reasons for discrepancies. Second, the comparison between the extrapolation and interpolation accuracy of TransGrow. Third, the flexibility in varying the sequence length at test time with $I_{\text{in}} = 1/3/5/7$ compared to training (fixed $I_{\text{in}} = 3$).

While we sample 200 times from each test sequence for Arabidopsis-S as in Sec. 6.3.2, we sample 20 times for GrowliFlower and Mixed-CKA because these two have shorter but more test sequences overall. The position of the target image within the sampled sequence I is fully random so that both interpolations and extrapolations are generated.

Quantitative results

Tab. 6.2 shows the interpolation and extrapolation scores of TransGrow for the datasets Arabidopsis-S, GrowliFlower and Mixed-CKA. In the interpolation with $I_{\text{in}} = 3$, Arabidospis-S has the highest image quality with MS-SSIM $= 0.9$ and

Table 6.2: TransGrow evaluation scores MS-SSIM, PSNR [dB], and FID for random temporal image generation of the datasets Arabidospis-S, GrowliFlower, and Mixed-CKA. Scores are divided in interpolation and extrapolation on test sequences using each $I_{\text{in}} = 1/3/5/7$ random input images.

Data	I_{in}	Interpolation		Extrapolation		FID (↓)
		MS-SSIM (↑)	PSNR (↑)	MS-SSIM (↑)	PSNR (↑)	
Arabidopsis-S	1	-	-	0.84 ± 0.10	27.90 ± 3.15	15.55
	3	0.90 ± 0.05	29.19 ± 2.29	0.86 ± 0.10	28.56 ± 3.43	14.83
	5	0.91 ± 0.05	29.51 ± 2.28	0.89 ± 0.08	29.35 ± 3.29	15.16
	7	0.91 ± 0.05	29.53 ± 2.25	0.88 ± 0.09	29.10 ± 3.68	15.56
GrowliFlower	1	-	-	0.23 ± 0.17	15.83 ± 3.99	256.07
	3	0.28 ± 0.16	16.23 ± 3.91	0.27 ± 0.17	16.34 ± 4.27	241.13
	5	0.29 ± 0.16	16.43 ± 3.89	0.29 ± 0.17	16.49 ± 4.50	235.42
	7	0.31 ± 0.17	16.85 ± 4.01	0.30 ± 0.16	16.33 ± 4.49	234.07
Mixed-CKA	1	-	-	0.22 ± 0.11	14.80 ± 1.93	91.57
	3	0.25 ± 0.07	15.08 ± 1.85	0.27 ± 0.13	15.23 ± 2.00	94.28
	5	0.26 ± 0.08	15.05 ± 1.92	0.26 ± 0.15	15.24 ± 2.13	96.95
	7	0.26 ± 0.09	15.36 ± 2.07	0.28 ± 0.16	15.36 ± 2.14	104.24

PSNR = 29.19, followed by GrowliFlower (MS-SSIM = 0.28, PSNR = 16.23) and Mixed-CKA (MS-SSIM = 0.25, PSNR = 15.08). Thus, there is a large difference in quality between the generated images of the different datasets, which is mainly due to the greater complexity of GrowliFlower and Mixed-CKA. Due to the larger time intervals, the changing backgrounds over time, the more diverse plant structures with several scattered plants per image, and the differences in the spectral image properties, it is significantly more challenging to generate an image that exactly matches the respective reference. The distribution of the generated images also does not match the reference distribution well, as indicated by the FID values of 241.13 for GrowliFlower and 94.28 for Mixed-CKA, although the qualitative results in both datasets show quite realistic plants.

Comparing the MS-SSIM of interpolation and extrapolation shows an average accuracy drop of 0.03 for Arabidopsis-S, 0.01 for Growliflower, and even a slight increase of 0.01 on average for Mixed-CKA. For Arabidopsis-S and Mixed-CKA, a large increase in the standard deviation can be seen. Overall, it can be said that the extrapolation accuracy is at the same level or just slightly lower than the interpolation accuracy. This is notable as the baseline methods comparatively do not allow an intuitive extrapolation.

By varying I_{in}, we aim to investigate how flexibly TransGrow adapts to different sequence lengths during inference, in particular those that deviate from the training sequence length $I_{\text{in}} = 3$. Across all datasets, an increase in I_{in} leads to

higher MS-SSIM and PSNR scores, whereby this increase from the shortest to the longest sequence is similar for interpolation and extrapolation ($\sim$ MS-SSIM +0.03 and PSNR +0.5). Particularly remarkable are the results for $I_{\text{in}} = 1$, i.e., the generation of time-varying images from a single input image. While baseline methods usually fail to generate those images, as the direction of movement in the latent space is ambiguous, this becomes feasible using explicit access (growth stage requesting in the input). The drop in accuracy to the sequence length $I_{\text{in}} = 3$ is low with an average MS-SSIM and PSNR change of -0.04 and -0.5, respectively. In general, the increase in accuracy through the addition of more support points is due to the greater information gain with simultaneously shorter interpolation and extrapolation distances. The sequence length difference between training and inference poses no problems, which demonstrates the flexibility of TransGrow.

Qualitative results and visualized variability

In the bottom nine rows of Fig. 6.3, the TransGrow interpolation and extrapolation results for Arabidopsis-S, GrowliFlower, and Mixed-CKA are shown, as well as the corresponding variability and reference images. While Arabidopsis-S achieves a high image quality, GrowliFlower and Mixed-CKA are significantly worse. With GrowliFlower, the leaf edges are blurred, and the images are not sharp overall. The Mixed-CKA images have better visual quality than GrowliFlower, and the wheat ears, in particular, are very sharp. However, the bean plants in between are rather blotchy, and their leaves are not well resolved. On the positive side, all datasets show consistency over time. All plants develop organically, the plant center positions remain correct over time, which is particularly evident in the early stages of Mixed-CKA, and no new plant parts appear in unnatural locations. Notably, the different color tones of the soil are visually accurately generated despite the very different conditions and the large deviations from the input images.

Focusing on the variability images, GrowliFlower shows that, similar to Arabidopsis-S, most of the variability is on the leaf edges, which is realistic. In addition, the soil is variable due to different soil moisture levels, smaller grasses, weeds, and furrows, which are more or less visible. The variability images of Mixed-CKA are less meaningful. Although the six rows of plants can also be identified here, which indicates that the variability lies mainly on the plant surfaces, the leaves are too mixed to be able to recognize differences between leaf margins and leaf centers. There are also no significant variability differences between wheat and bean plants. There are points in time with greater variability, such as 99 DAS, and other points in time with less variability, such as 74 DAS. It was expected that the images generated at the sampling points, in particular, would have lower variability. However, this is not the case for all datasets, which

could be due to TransGrow not being trained as an identity mapper, so instead, the data variability within a time point is noticeable.

A particular focus is on the generation of extrapolation images. In the case of Arabidopsis-S, it is especially clear that the plant becomes smaller when extrapolating into the past and that the outer leaves gradually expand outwards and become wider during the two extrapolation steps into the future. Similarly, the extrapolations of GrowliFlower and Mixed-CKA are highly realistic. Smaller plants (GrowliFlower) or bare soil (Mixed-CKA) can be seen for the first point in time, and fully developed cauliflower heads or a ripened mixture field for the last two points in time. It is conceivable that such time-flexible extrapolations into the harvest period can provide high-added value for agricultural practice.

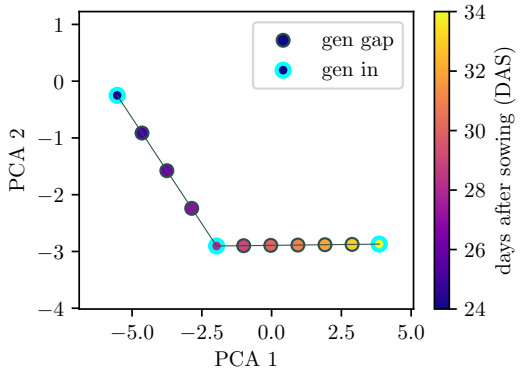
6.3.4 Visualizing inter- and extrapolations in the latent space

To analyze how the latent code of a generated plant evolves, we consider in detail the transformed pred-token, which can be seen as the latent representation of the generated image X_{gen}. Ideally, the generated images at consecutive growth stages do not scatter randomly but follow a pattern in the latent space, which is referred to as latent trajectory. Note that for each latent trajectory, the stochasticity is kept constant, and only the requested time is varied. To visualize this, the generated latent codes of pred-tokens from multiple requested times are saved, and together with Principal Component Analysis (PCA) [141] reduced to two dimensions. We also explored other dimensionality reduction methods, particularly for extracting the manifold of the latent space, but PCA produces the most intuitive visualization.

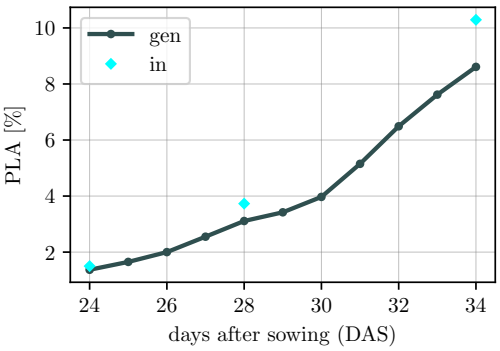
In Fig. 6.5, linear interpolation of pred-tokens in the TransGrow latent space is compared with the target time requesting applied in TransGrow. For the latter, trajectories generated with two different noise vectors are shown. The PLA is calculated from the generated images for both methods to visualize the effects of different latent space methods on the image space.

There are clear differences between the methods in the latent space. While with target time request, the trajectories of ϵ_0 and ϵ_1 between 24 DAS and 28 DAS are still close to the linear interpolation, between 28 and 34 DAS, they show a significant difference to the linear interpolation. This is visible by a slightly larger development step between 32 DAS and 33 DAS, both in the PLA and in the generated images. Both possibilities are realistic scenarios, as plant growth is not linear.

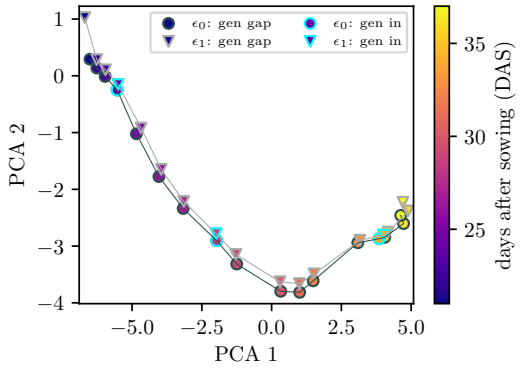
The target time request enables extrapolation whose points in the latent space do not lie on the linear extension between the nearest interpolation points. How-



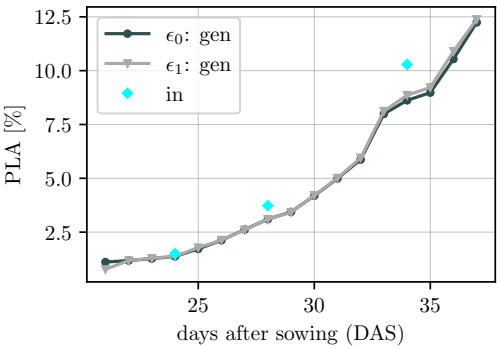
(a) Latent space visualization: Latent interpolation



(b) PLA development: Latent interpolation



(c) Latent space visualization: Target time request



(d) PLA development: Target time request

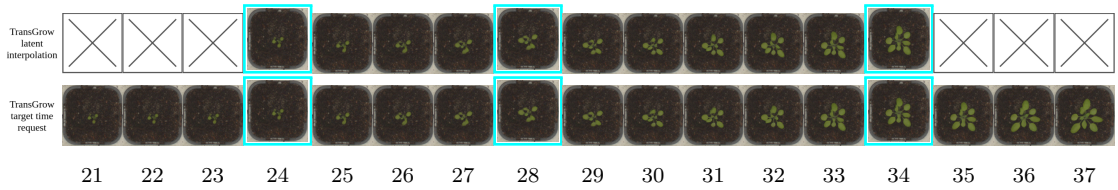


Figure 6.5: Visualization of the latent space of the transformed pred-token in image generations over the time of $I_{\text{in}} = 3$ constant input images. Comparison of (a) latent interpolation with (c) target time request in TransGrow and (b)+(d) the respective effects on the projected leaf area (PLA) development. With the target time request, the images were generated with different noise $\boldsymbol{\epsilon}_0$ and $\boldsymbol{\epsilon}_1$, but constant over time. At the bottom, the qualitative results are shown for latent interpolation and target time request using $\boldsymbol{\epsilon}_0$.

ever, the image space and the PLA development confirm a realistic extrapolation. The difference between ϵ_0 and ϵ_1 shows that although the points in the latent space are slightly apart, measurable differences in terms of PLA are only present in the extrapolation range (21 DAS and 34-37 DAS). It can be seen from the extrapolation in the future that the distance in the visualized latent space does not correlate directly with changes in the image space. Here, the latent space points are very close to each other, but significant changes in plant development take place in the image space.

There are possibilities to regularize the latent space with additional loss functions, e.g., so that the temporal distance correlates with the trajectory length in the latent space. However, such a regularization would have to take into account the not-exactly known and non-linear plant development, which makes it challenging to implement. For instance, the difference in the latent space would be expected to be significantly smaller between 21 DAS and 22 DAS than between 36 DAS and 37 DAS.

In general, the greater the distance between the sampling points, the greater the risk of leaving the data manifold during linear interpolation and thus generating unnatural images, which is also a disadvantage of linear interpolation within VAE and AAE. In our experiment, both methods provide different latent space visualizations but realistic generated images, with the advantage of TransGrow, which makes realistic extrapolations feasible.

6.4 Discussion

6.4.1 On interpolation and extrapolation in latent space

Linear interpolation in the latent space typically results in a straight line between two points. However, the actual distribution of data in the latent space might not be linear. It can be curved or twisted, meaning that the shortest path between two points might not necessarily be a straight line. More sophisticated techniques, such as geodesic or cubic B-Spline interpolation, attempt to find shorter paths along the manifold connecting two or more points in the latent space [139]. However, these methods require knowledge or assumptions about the shape of the manifold and a choice of distance metric, and the calculation is computationally intense depending on the manifold.

Latent extrapolation is also conceivable by extrapolating the last two or more sampling points, but this has not led to plausible results in prior experiments. The locations of the support points compared to the extrapolated images in the visualized latent space (Fig. 6.5) provide an indication of why linear extrapolation often leads to unrealistic images.

In summary, linear interpolation, the most intuitive and simple technique, is used in this chapter to primarily visualize the differences to explicit latent space access via the requested growth stage. While it is not precluded that more advanced interpolation techniques may provide better interpolation results, extrapolation remains challenging.

6.4.2 Flexibility of the TransGrow framework

Benefits of flexibility

As the experiments show, TransGrow has great flexibility in that irregular sequences with different lengths and time intervals can be included in the input, and any growth stages can be generated. This variability is not only practical for agricultural practice, e.g., adaptively adding images of new drone overflights to an existing model, but can also be useful in related applications. For instance, in earth observation using satellite image time series, where some points in time are unusable due to clouds and can be reconstructed in a generative manner using the remaining, thus irregular, observations.

It should also be emphasized that the positional encoding can vary the time unit of the model, which in our experiment is fixed to day increments. In this way, the same data can also be used to train models that can, for instance, interpolate and extrapolate on an hourly basis, i.e., in much smaller increments, if required.

Drawbacks of flexibility

There are two main criticisms: First, the results have shown that the closest image in time has the greatest influence when generating the target image. While this is reasonable, the addition of more support points would be expected to result in a more significant increase, as the model should be able to better capture the overall growth behavior of the plant over time. The fact that very realistic images are already generated with one input image contradicts this assumption.

Second, the identity mappings, i.e., images generated at time points of the input images with TransGrow, are of poorer quality than the comparison methods. In addition, they have a high standard deviation for different stochastic components, which is not reasonable because only the input image of the correct point in time should be reconstructed at these points in time. One explanation is that VAE and AAE were trained as identity mappers, while TransGrow always has an input sequence. If the image to be generated exists in the input, TransGrow does not succeed in completely suppressing the other images of the input sequence, which would be desirable. Additional experiments with the aim of improving identity mapping by reconstructing support points more frequently during training or controlling the stochastic component depending on the temporal distance to the target image (less noise the smaller $\min \Delta \boldsymbol{t}$) did not solve the issue.

This means that TransGrow's flexibility to input several images simultaneously comes at the expense of error-free identity mapping. The next chapter addresses both drawbacks through flexible image generation but with only one input image at a time.

6.4.3 Comparison to image-to-image translation

Compared to data-driven growth modeling using image-to-image translation (Chap. 5), TransGrow has several advantages. In addition to the flexibility already mentioned, whereby arbitrary points in time can be generated instead of a predefined growth step, the greater diversity of the output, including the variability images, is a significant achievement. By optimizing using CWGAN instead of classical CGAN, it is possible to prevent mode collapse on the one hand and not to suppress the stochastic components $\boldsymbol{\epsilon}$ on the other. This allows variability images to be generated, while the image-to-image in Chap. 5 completely suppresses the stochastic component in the input.

Explanations for differences in image quality

Whereas the Arabidopsis-S dataset is not comparable to the Arabidopsis-P dataset, in part due to the different augmentation applied, the datasets from real field environments can be compared: While image-to-image translation with Brassica resulted in an FID score of 30 - 40 depending on the treatment, the FID for GrowliFlower (>230) and Mixed-CKA (>90) are significantly higher. Although the datasets are not more complex, worse image quality is achieved with TransGrow.

This is attributable to three factors: The first responsibility comes from the flexibility of TransGrow. Focusing on individual output time points is much easier for the model, as in image-to-image translation, where the modeled growth prediction step is always identical. In TransGrow experiments, where the target image was fixed to a specific position in the input sequence during training, the FID could be increased by up to 50 %. However, in this way, it loses the ability to generate time-varying images during inference.

Second, the smaller latent space size is crucial. While TransGrow's latent space has a size of $d_{\text{model}} = 512$, the paired imagine-to-image translation uses skip connections between the encoder and decoder, which means that far more information can be transferred from the input to the image to be generated. However, the smaller latent space of TransGrow is necessary and is caused by the processing of spatio-temporal embeddings using a transformer. Due to this architecture, no skip-connections are possible, and the parameter-intensive attention mechanism within the transformer does not allow a significantly larger latent space for reasons of training time. Nevertheless, a latent space with a size of $d_{\text{model}} = 512$ can also be sufficient for high image quality, as related work demonstrates (Sec. 6.1).

This is where a third factor takes effect: Large training datasets are required to train transformers. This size could be artificially generated for Arabidopsis-S through the special shuffling augmentation but not for GrowliFlower and Mixed-

CKA.

In summary, TransGrow's image quality can be improved by lower output flexibility, a larger latent space combined with more parameter- and runtime-intensive training, and larger and more diverse datasets.

6.5 Conclusion

In this chapter, we have shown that our conditional Wasserstein generative adversarial network TransGrow, with a combined generator of convolutional neural networks and transformers, enables high-quality and realistic image generation for incomplete and irregular sequences. For the three datasets, Arabidopsis-S, GrowliFlower, and Mixed-CKA, time-dependent sampling in the latent space ensures a substantial reduction of Fréchet inception distance compared to variational and adversarial autoencoder approaches utilizing linear latent space interpolation. We investigated the accuracy of the interpolation in dependence on the distance to the nearest support point and found that although TransGrow has difficulties with identity mappings, it can perform significantly better long-term interpolations. Visualizing the latent space of pred-tokens over time using dimension reductions can indicate potential reasons: For short interpolation distances, only small differences in the latent space between linear interpolation and target time requesting are visible. Therefore, the differences are also minor in the image space and in the projected leaf area derived from the generated images. However, for larger interpolation distances, we observe that linear interpolation deviates from the data manifold, while TransGrow still samples from it

In addition to interpolation, TransGrow is also capable of extrapolation with only a minimal decrease in multiscale structural similarity and peak signal-to-noise ratio. Furthermore, pixel-wise variability images for each time can be derived from a generated output image distribution, indicating reliable pronounced variance at the leaf edges, where the variance of plant growth is naturally highest. This allows farmers to forecast multiple future probable above-ground phenotypes from a flexible number of images and any time points in the growing season.

Chapter 7

Multi-modal conditional image generation and simulation

This chapter is about combining a CGM based on images with other conditions, i.e. growth influencing factors of different kinds, to be able to perform multi-modal conditional image generation. Special emphasis is placed on the variation of these growth influencing factors for inference to perform growth simulations.

In this context, we aim to incorporate the strengths of the CGMs from the previous chapters. On the one hand, the less complex model architecture with only one input and output image at a time is to be retained, which has led to high image quality, as in Chap. 5. On the other hand, we want to keep the flexibility to choose the input and output times of the images arbitrarily, coupled with the ability to generate realistic output distributions, as in Chap. 6. Linking both together with multi-modal conditioning, which we implement using conditional batch normalization to realistically consider other growth influencing factors alongside the input image, leads to a growth simulation framework as shown in Fig. 7.1.

It is a two-step procedure in which time-varying images are first generated with the IGM and then analyzed with an independently trained GEM. An important novelty in the IGM, which is a CWGAN, is the integration of multiple conditions of different types. These are images (2D spatial continuous variables), time points (discrete), treatment information (categorical), and daily simulated biomass (continuous). This enables simulations during inference, i.e., while fixing input conditions (input image, time, and treatment), for other growth stages, conditions can be varied as required to generate multiple realistic predictions, as shown with the gray shaded images in Fig. 7.1.

Experiments have been conducted on different datasets of varying complexity, from Arabidopsis-S to real field data with cauliflower (GrowliFlower) and crop mixtures (MixedCrop). In addition to classical GAN evaluation metrics,

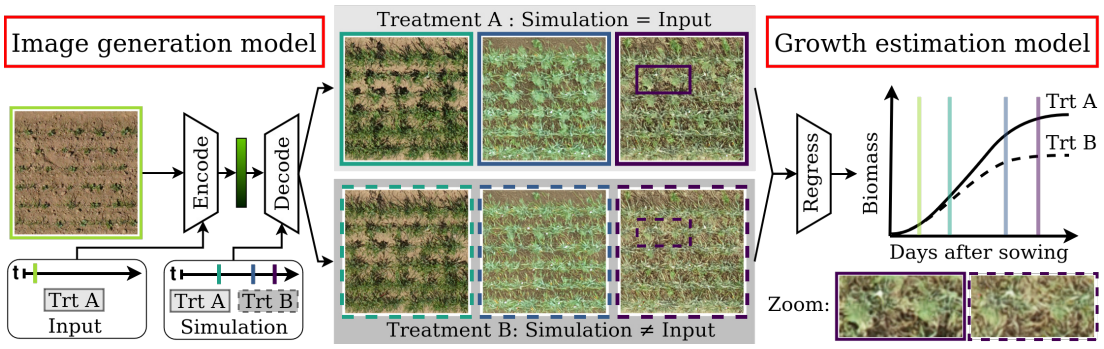


Figure 7.1: Proposed two-step crop growth simulation framework: In the first step of image generation, an input image is initially encoded with its associated time (t) and treatment (trt). Then, this encoded representation can be decoded into newly generated images with varying growth stages for different simulation times and treatments. In the second step of growth estimation, target parameters such as projected leaf area or biomass are estimated from the images and analyzed over time. Both models are trained independently.

we evaluate the quality of generated images through the GEM, which acts as a plant phenotyping module, by comparing (depending on the dataset) either the projected leaf area or the biomass estimated from generated and real images.

It is worth emphasizing that the biomass condition for the MixedCrop dataset is not a real growth influencing factor but instead simulated using a process-based CGM. For crop mixtures, this allows us to make a comparison between our image-based crop growth simulation and a classic process-based one, which was used to establish the GEM. Thereby, we demonstrate that the IGM can serve as an interface that makes the output of process-based CGMs more explainable by visualizing the spatial crop development.

A transferability experiment demonstrates that our framework has the potential to be transferred to crop mixtures in another field with different environmental conditions. Finally, we also investigate the generalizability and transferability of our framework from a temporal and spatial perspective: Temporally, by generating images at times that do not exist as acquisition times in the training dataset. Spatially (and temporally), by applying the crop mixture CGM trained on Mixed-CKA to Mixed-WG images (with training-different acquisition times).

The primary new findings of this chapter can be outlined as follows:

- Multi-modal growth influencing factors are integrated as conditions into an image generation model using conditional batch normalization and allow the data-driven CGM to perform realistic growth simulations.
- Generated images deviate less from the appearance of the reference plant the more conditions are integrated, i.e. the more precisely the growth be-

havior is described.

- By incorporating process-based simulated biomass as a condition, a serial interface between PBM and DDM is created, which can be used to obtain for PBM additional spatial explainability, validation, or indications of re-calibration for certain field treatments.

7.1 State of the art

In many data-driven image generation works, plant growth is greatly simplified by considering only a few conditions as growth influencing factors, for instance, the time factor [15], or shape priors [94], while in fact, it is subject to many factors. Miranda et al. [20] attempt to get closer to this complexity by integrating more conditions into the growth modeling, which allows them to generate controlled and explainable output images of crop mixtures. In their work, the conditions are not limited to growth-influencing factors but also include regulating variables such as seed numbers, the mean plant height, or the plant's biomass. The conditioning works by replacing the stochastic input of a combined VAE GAN inside the BicycleGAN framework [142] with a conditional input. However, the proposed method is limited to a fixed number of continuous conditions and to a predefined growth prediction step from a fixed early growth stage to a fixed later growth stage, which is unfavorable in agricultural practice.

In general, integrating multiple conditions is a non-trivial task, as in conditional image generation, there is a trade-off between sample variety and fidelity [143]. When the model is optimized for high variety, it aims to produce a broad range of outputs for a given input condition, which leads to diverse outputs but images that are less accurately conditioned (lower fidelity). When the model is optimized for high fidelity, images are of good quality and accurately conditioned but become less diverse (lower variability). The latter can lead to the model generating completely deterministic outputs, which have a similar effect to mode collapse [144].

There are many different ways of integrating conditions from concatenation [111] over auxiliary classifiers [145], feature-wise linear modulation [146], adaptive instance normalization [124], and latent projection [147] to conditional batch normalization [143], [148]. In this chapter, conditional batch normalization is used since it allows the intuitive integration of multiple conditions while maintaining the stochasticity of the model to create an adequate distribution of generated plants. There are also other advantages, such as seamless integration into a fixed model architecture, whereas with classifier guidance, for example, an auxiliary classifier must be trained.

7.2 Methods

This section provides details of the framework[1] and its components, whereby a two-step approach is applied. First, an image is predicted (Sec. 7.2.1) using the Image Generation Model (IGM), and second, the growth is estimated (Sec. 7.2.2) using plant phenotyping within a Growth Estimation Model (GEM). While existing state-of-the-art models are used for growth estimation, which is fine-tuned on our data, the methodological focus is on integrating multiple conditions of different types in the architecture of the IGM to predict new images.

To specify the terminology: We call the output of the image generation model generated or predicted image. The whole framework's output is called data-driven prediction, in contrast to process-based prediction, which is process-based simulated biomass. In the case of predictions, there is a time shift $\Delta t = t_{\text{gen}} - t_{\text{in}}$, so $\Delta t > 0$ means prediction into the future, $\Delta t < 0$ means prediction into the past, and $\Delta t = 0$ is an identity mapping (T_0). The output of the growth estimation model is an estimation (no time shift) relating to its own input but a prediction relating to the input of the preceding image generation model.

7.2.1 Multi-modal conditional image generation

For image generation, we build a multi-conditional Wasserstein GAN with gradient penalty (CWGAN-GP) [35] from several state-of-the-art components. The network consists of a generator $\mathcal{G}_\theta$ and a discriminator $\mathcal{D}_\delta$, where $\mathcal{G}_\theta$ predicts images and $\mathcal{D}_\delta$ estimates a score for generated and real images.

Conditional Wasserstein GAN objective

In the generator, a target image $\boldsymbol{X}_{\text{gen}} = \mathcal{G}_\theta(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})$ is generated from an input image $\boldsymbol{X}_{\text{in}}$, further conditions $\boldsymbol{y}$, and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$. It should be noted that everything that is not $\boldsymbol{\epsilon}$ represents a condition according to the CGAN definition described in Sec. 2.4.2, i.e. also the input image $\boldsymbol{X}_{\text{in}}$. In this case, the notation separates it from the other conditions $\boldsymbol{y}$ since these can be different for the input image and the image to be generated and thus split into $[\boldsymbol{y}_{\text{in}}, \boldsymbol{y}_{\text{gen}}]$. Both $\boldsymbol{y}_{\text{in}}$ and $\boldsymbol{y}_{\text{gen}}$ represent multi-conditioning, which can be composed of several of the following conditions: categorical (class) variables c, discrete variables t, and continuous variables $\boldsymbol{b}$. In the discriminator, either the reference $\mathcal{D}_\delta(\boldsymbol{X}_{\text{ref}}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})$ or the generated image $\mathcal{D}_\delta(\boldsymbol{X}_{\text{gen}}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})$ are presented along with input image and all conditions. The discriminator estimates a score for both real and generated input, which is capable of enforcing the minimization of the Wasserstein distance between the two distributions. The objective of adversarial training is to optimize

[1] Source code is publicly available at `https://github.com/luked12/crop-growth-cgan`

the parameters θ and δ by maximizing the objective function $L_{\text{GAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ by $\mathcal{D}_\delta$ and minimizing it by $\mathcal{G}_\theta$.

$$\theta^*, \delta^* = \arg\min_\theta \arg\max_\delta \mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta) \tag{7.1}$$

Eq. 7.2 represents $\mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ with the classic CWGAN objective in the first two lines [34], added with the gradient penalty term in the second line to enforce the required 1-Lipschitz continuity of $\mathcal{D}_\delta$ [35].

$$\begin{aligned}\mathcal{L}_{\text{CWGAN}}(\theta, \delta; \boldsymbol{\epsilon}, \boldsymbol{X}_{\text{ref}}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y}) &= \mathbb{E}_{(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})}[\mathcal{D}_\delta(\mathcal{G}_\theta(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y}), \boldsymbol{X}_{\text{in}}, \boldsymbol{y})] \\ &\quad - \mathbb{E}_{(\boldsymbol{X}_{\text{ref}}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})}[\mathcal{D}_\delta(\boldsymbol{X}_{\text{ref}}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})] \\ &\quad + \lambda_{\text{GP}} \mathbb{E}_{(\boldsymbol{X}_{\text{in}}, \hat{\boldsymbol{X}})}[(\|\nabla_{\hat{\boldsymbol{X}}} \mathcal{D}_\delta(\boldsymbol{X}_{\text{in}}, \hat{\boldsymbol{X}})\|_2 - 1)^2]\end{aligned} \tag{7.2}$$

The gradient penalty is computed by blending a generated image with a reference image, resulting in $\hat{\boldsymbol{X}} = \eta \boldsymbol{X}_{\text{ref}} + (1 - \eta)\mathcal{G}_\theta(\boldsymbol{\epsilon}, \boldsymbol{X}_{\text{in}}, \boldsymbol{y})$, where η is a random value in the range [0, 1], and its impact is controlled by λ_{GP}. Using $\mathcal{L}_{\text{CWGAN}}(\mathcal{G}_\theta, \mathcal{D}_\delta)$ minimizes the Wasserstein-1 distance, sidestepping issues like mode collapse and vanishing gradients in classic GAN training.

Network architecture with multi-conditioning

Generator. The generator consists of an encoder $\mathcal{Q}$ that compresses the input image and conditions related to the input image into a latent representation $\boldsymbol{z} = \mathcal{Q}(\boldsymbol{X}_{\text{in}}, \boldsymbol{y}_{\text{in}})$ and a decoder $\mathcal{P}$ that generates the target image from a stochastic component, the latent representation, and the conditions for the image to be generated $\boldsymbol{X}_{\text{gen}} = \mathcal{P}(\boldsymbol{\epsilon}, \boldsymbol{z}, \boldsymbol{y}_{\text{gen}})$. While for image encoding, a ResNet-18 backbone [42] without a final fully connected layer and global average pooling with pre-trained ImageNet [134] weights is used, decoding works architecturally inverse to that. To integrate the conditions, all batch normalization layers are replaced by conditional batch normalization layers (CBN) [149], where the learnable affine parameters of classical batch normalization layers [150] are conditioned on some auxiliary variable $\boldsymbol{a}$. In our case, $\boldsymbol{a}$ are embeddings of the conditions $\boldsymbol{y}$ using an embedding function Φ. In particular, the encoder's CBN layers are conditioned on the embeddings related to the input image $\boldsymbol{a}_{\text{in}} = \Phi(\boldsymbol{y}_{\text{in}})$, while the decoder's CBN layers are conditioned on the embeddings related to the image to be generated $\boldsymbol{a}_{\text{gen}} = \Phi(\boldsymbol{y}_{\text{gen}})$. Specifically, the embedding function is condition-type-specific since $\boldsymbol{y}$ can consist of conditions of up to 3 different types, namely discrete temporal information t, categorical class information c, and continuous variables $\boldsymbol{b}$. So individual embeddings are performed for each type of condition in $\boldsymbol{y}$, which are then concatenated to $\boldsymbol{a}$.

$$\begin{aligned}\boldsymbol{y}_{\text{in}} &= [t_{\text{in}}, c_{\text{in}}, \boldsymbol{b}_{\text{in}}], & \boldsymbol{y}_{\text{gen}} &= [t_{\text{gen}}, c_{\text{gen}}, \boldsymbol{b}_{\text{gen}}] \\ \boldsymbol{a}_{\text{in}} &= [\Phi_t(t_{\text{in}}), \Phi_c(c_{\text{in}}), \Phi_b(\boldsymbol{b}_{\text{in}})], & \boldsymbol{a}_{\text{gen}} &= [\Phi_t(t_{\text{gen}}), \Phi_c(c_{\text{gen}}), \Phi_b(\boldsymbol{b}_{\text{gen}})]\end{aligned} \tag{7.3}$$

Here, the temporal embedding Φ_t consists of positional encoding of discrete time points followed by a two-layer MLP with a sigmoid linear unit (SiLU) function in between. The class embedding Φ_c represents a classic lookup table embedding that maps indices of categorical class variables to a continuous vector representation. In order to embed a vector of continuous values in Φ_b, a two-layer MLP with SiLU function in between is used. In the experiments, the conditions c and $\boldsymbol{b}$ are not always used, then embedding and resp. concatenating of unused conditions is omitted. Notably, for the MixedCrop dataset, $t_{\text{in}}/t_{\text{gen}}$ and $c_{\text{in}}/c_{\text{gen}}$ are scalars representing time (t) and treatment (c), respectively, while $\boldsymbol{b}_{\text{in}}/\boldsymbol{b}_{\text{gen}}$ are vectors representing 2-dimensional due to SW and FB biomass. However, after embedding the individual components of $\boldsymbol{y}$, it is ensured that $\Phi_t(\boldsymbol{t})$, $\Phi_c(\boldsymbol{c})$, and $\Phi_a(\boldsymbol{b})$ all represent continuous vectors of the same 64-dimensional embedding size, which avoid prior weighting of different conditions. Besides, CBN has already included a linear embedding for all conditions, but the additional condition-type-specific embedding has stabilized the training process.

To also incorporate stochasticity into the network, a random 128-dim noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1) \in \mathcal{E}$ is generated and via noise mapping network $f : \mathcal{E} \mapsto \mathcal{W}$ inspired by StyleGAN [124] projected to the latent code $\boldsymbol{w} \in \mathcal{W}$, that matches the channel dimension of the latent representation $\boldsymbol{z}$. The mapping network f is a shallow three-layer linear embedding network, which gradually projects the 128-dimensional $\boldsymbol{\epsilon}$ to the 512-dimensional $\boldsymbol{w}$, which corresponds to the channel size of the ResNet-18 latent representation. After repeating $\boldsymbol{w}$ for the spatial dimension (global average pooling is omitted), it is finally added to $\boldsymbol{z}$. This means that the stochasticity is not incorporated in the encoder part of the generator but is only used in the decoder part.

Discriminator. The discriminator takes either the generated $\boldsymbol{X}_{\text{gen}}$ or reference image $\boldsymbol{X}_{\text{ref}}$ along with the input image $\boldsymbol{X}_{\text{in}}$, and the conditions $\boldsymbol{y}$ as input. The images are concatenated channel-wise in the input and initially passed through a convolutional layer and LeakyReLU activation. This is followed by several convolutional blocks consisting of a convolutional layer, instance normalization, and LeakyReLU up to a spatial dimension of $[16 \times 16]$. Since batch normalization should be avoided in the Wasserstein discriminator [35], the conditions are not integrated in this case with conditional batch normalization. Instead, each condition is first embedded to dimension 256 with a different embedding function Ψ than Φ in the generator, but the architecture of the embedding functions inside Ψ and Φ are the same. Then, embedded conditions are reshaped and channel-wise concatenated to the intermediate discriminator representation of spatial size $[16 \times 16]$. Note that here, the conditions of both the input image and the image to be generated are concatenated. From this concatenated representation, the

final score is generated with further convolutional blocks. Previous experiments have shown that the training converges significantly better with an intermediate fusion of the conditions than with a fusion directly in the discriminator input.

Optimization by random sampling of image pairs

The data sampling is special since multiple reference images can be used for every input image due to the possibility of temporal conditioning. Thus, we use a sampling approach similar to TransGrow with random data sampling Sec. 6.2.1. In each epoch, we first iterate classically over all training images, which are then used as input images. Second, always another random image of the same plant is sampled for each input image, representing the reference plant and completing the image pair used for training. The conditions $\boldsymbol{y}_{\text{in}}$ and $\boldsymbol{y}_{\text{gen}}$ are drawn according to the sampled images. This causes that during the training $c_{\text{in}}{=}c_{\text{gen}}$ because the treatment class does not change over time. The sampling procedure for calculating test scores is identical. Each test image represents an input image once and is assigned a random growth stage as the reference image to be generated. For inference, the conditions can be varied arbitrarily, what we call data-driven simulation. So a treatment change $c_{\text{in}} \neq c_{\text{gen}}$ is possible, $\boldsymbol{b}$ does not have to fit the reference values, and t can deviate from the training range.

7.2.2 Evaluation of generated images

Evaluation of image quality

To evaluate the quality of the generated images, we use a well-established set of GAN evaluation metrics. For the direct comparison between generated and reference images of the same time point, we use the Multi-scale Structural Similarity Index Measure (MS-SSIM [39], optimal: 1) and the Learned Perceptual Image Patch Similarity (LPIPS [37], optimal: 0). While MS-SSIM compares the generated with the reference image directly at different resolutions of the image space, LPIPS evaluates the similarity of image patch activations in the VGG-embedded latent space, which has been shown to have a high correlation with human perception. In addition, the Fréchet Inception Distance (FID [38], optimal: 0) is used to compare not only the quality but also the diversity of the generated image distribution with the real image distribution of the test dataset. In contrast to Sec. 5.2.2, only the classic $\text{FID}(\mathcal{N}_g, \mathcal{N}_r)$ is used. However, for long-term predictions far into the future or past, that means a large difference exists in the growth stage of the input image and the image to be generated, so it is not expected that generated and reference images match at the pixel level. Although FID will degrade less as long as the plants fit into the distribution of each growth stage, poor results are to be expected for MS-SSIM and LPIPS in such cases. To

evaluate whether useful plant-related traits can still be derived, we use GEMs, which determines leaf area and biomass from the generated images, as described below.

Growth estimation by projected leaf area

For Arabidopsis-S and GrowliFlower, growth is determined using the plant trait projected leaf area (PLA). Both datasets are well suited for this purpose because different plants do not overlap until advanced growth stages. The PLA is derived as an image-wise pixel sum of plant segmentations predicted with a Mask R-CNN instance segmentation model [43]. For this, two models, with pre-trained ImageNet weights [151], are fine-tuned on a few images of the respective plant dataset, for which reference segmentation masks are available. By multiplying the PLA with the squared dataset-dependent ground sample distance (GSD), we report PLA in the unit mm^2 for Arabidopsis-S and cm^2 for GrowliFlower or for comparability normalized in the unit [% image^{-1}], which is achieved by dividing the PLA by the image size. In this chapter, PLA is not calculated for the whole image but only out of the segmentation predictions for the center plant, which is especially relevant for GrowliFlower, where there are, in most cases, multiple plants per image. To compare the PLA of a single generated and reference image pair, we use $\Delta\mathrm{PLA} = \mathrm{PLA}^{\mathrm{gen}} - \mathrm{PLA}^{\mathrm{ref}}$. For MixedCrop, PLA cannot be extracted with sufficient accuracy at the pixel level for the individual crop species due to the fine structure of the wheat ears, enormous plant overlap, and a lack of annotated images [19]. The accuracy evaluation of the trained instance segmentation models can be found in Sec. 7.3.2.

Growth estimation by biomass

Instead of PLA, for MixedCrop, dried biomass (BM) in tons per hectare [$\mathrm{t\,ha}^{-1}$] is to be derived from the images as a growth indicator, divided into the two mixture species spring wheat (SW) and faba bean (FB). To estimate both with one model, a ResNet-18 [42] is used, modifying the last linear layer to two output neurons, which are activated with ReLU, since only positive biomass values are possible. The mean squared error (MSE) function is used as the loss function. We use weights from a pre-training with ImageNet [151] and fine-tune on MixedCrop images and corresponding reference biomass values. These reference biomass values are not actual in-field measurements but come from a process-based CGM for mixtures (see Sec. 4.2.4) that provides simulated SW and FB biomasses dynamically for each image time point. Notably, we use the same simulated biomass values that are used as conditions in the image generation part of the framework. However, this dual use is methodologically not critical since the

image generation part and the growth estimation part are trained independently of each other. Similar to PLA, we use $\Delta\text{BM} = \text{BM}^{\text{gen}} - \text{BM}^{\text{ref}}$ to report biomass deviations between two images. Overall, estimating biomass from bird's eye view imagery has three main challenges and inherent sources of error. First, biomass is a 3D quantity derived from 2D images. Second, the process-based CGM only estimates dried biomass for all growth stages, which is used as a reference for training the GEM. However, the images show plants with their actual humidity (fresh matter), which changes over time. Third, the simulation result varies only treatment-wise, but it is likely that plants of the same treatment will develop differently in multiple replications in the field due to different soil conditions. For the discussion about the biomass estimation results and accuracies, see Sec. 7.3.2.

7.3 Experiments and results

In this section, after the experimental setup, the results of the GEMs are described at the beginning, as the accuracies of these models are needed for the discussion of the image generation results. In the following, we first show the results of image generation with only temporal variation, which allows a comparison with reference data, then simulations with further changed conditions, and finally, the transferability to another experimental site.

7.3.1 Experimental setup

Data augmentation and preparation

As image augmentations, horizontal and vertical flipping, 90° rotations, slight translations within a random affine transformation, and ShadowOut, which is a semi-transparent version of CutOut [152], are applied simultaneously to input and reference or generated image. Using a single NVIDIA A100-PCIE-40GB and a batch size of 64, the training duration is between 13 d and 35 d, depending on the dataset size.

Model hyperparameter

Adam optimizer is used with a learning rate of 1e-4 for both $\mathcal{G}_\theta$ and $\mathcal{D}_\delta$ optimization. Regardless of the number of conditions, the models are trained for 5000 epochs, after which the best epoch is selected based on the lowest LPIPS on the validation data.

Table 7.1: Mask R-CNN instance segmentation accuracies divided into bounding box and segmentation for the real (non-generated) images of the test set. Overall average precision (AP), with thresholds at IoU = 0.50 and IoU = 0.75, and overall average recall (AR) are given.

	Bounding Box				Segmentation			
	AP ø	AP @0.50	AP @0.75	AR ø	AP ø	AP @0.50	AP @0.75	AR ø
Arabidopsis-S	0.92	0.99	0.99	0.95	0.77	0.99	0.98	0.78
GrowliFlower	0.86	0.96	0.92	0.88	0.78	0.97	0.92	0.82

Table 7.2: Biomass estimation accuracies assessed by MAE and ME between the estimations from the real (non-generated) images of the test set and reference values from the process-based crop growth model. In addition to an overall (OA) score, scores are calculated separately for mixtures and SW resp. FB monocultural fields. All units are given in $\mathrm{t\,ha^{-1}}$.

		Mixtures		$\mathrm{SW_{mono}}$		$\mathrm{FB_{mono}}$		OA	
		MAE	ME	MAE	ME	MAE	ME	MAE	ME
CKA	SW	0.142	-0.006	0.188	-0.074	0.001	0.001	0.142	-0.026
	FB	0.125	-0.008	0.017	0.017	0.179	0.052	0.097	0.005
WG	SW	0.126	-0.023	0.150	-0.050	0.018	0.018	0.122	-0.027
	FB	0.105	-0.026	0.003	0.003	0.185	-0.046	0.082	-0.019

7.3.2 Accuracy assessment of growth estimation models

Accuracy of projected leaf area estimation

Instance segmentation, which is used to derive PLA (projected leaf area), is trained on a small subset of the corresponding datasets for which reference segmentation masks are available. Exact numbers for all datasets can be found in the bottom part of Tab. 4.3. The reference masks of the test set specified there are used to run the evaluation in Tab. 7.1. It shows the instance segmentation accuracies using the measures AP and AR, which - due to their direct derivation from these - correlate with the accuracy of the PLA. The GrowliFlower accuracies are comparable to the results of Kierdorf et al. [52], i.e., sufficient to evaluate cauliflower growth. Arabidopsis-S has a higher AP and AR for bounding boxes and is at a comparable high level to GrowliFlower for segmentation, thus also adequate to determine PLA.

Accuracy of biomass estimation

The accuracy of dried biomass estimation for both MixedCrop sites is given in Tab. 7.2. For mixtures the MAE is between $0.126\,\mathrm{t\,ha^{-1}}$ and $0.142\,\mathrm{t\,ha^{-1}}$ for

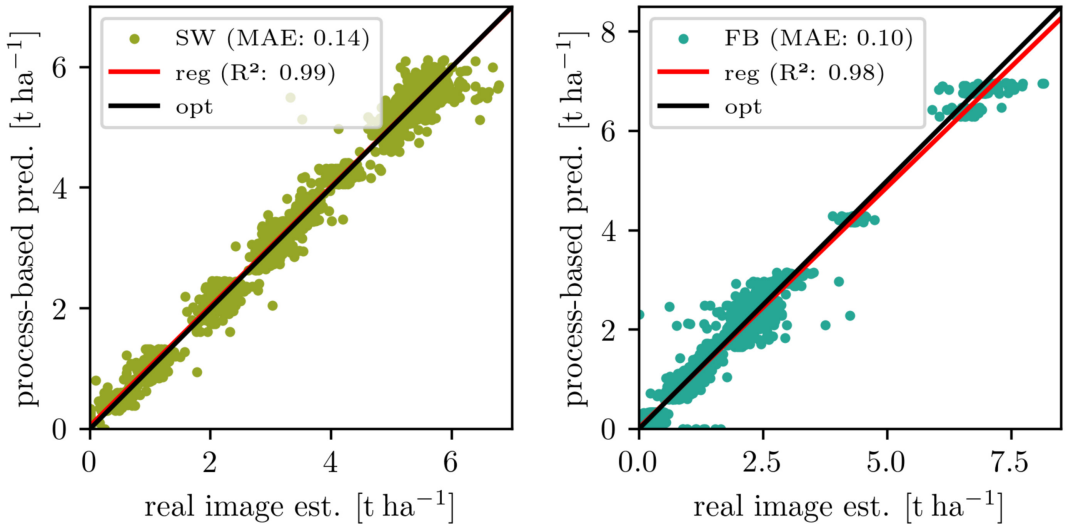


Figure 7.2: Scatter results of dried biomass estimation from real Mixed-CKA imagery overall growth stages and treatments (mixtures and monocultural fields) split up in spring wheat (SW) and faba bean (FB). The process-based predictions are used as a reference. The regression line is shown in red, and the optimal line is in black.

SW and between $0.105\,\mathrm{t\,ha^{-1}}$ and $0.125\,\mathrm{t\,ha^{-1}}$ for FB. Notably, the ME is less than $-0.01\,\mathrm{t\,ha^{-1}}$ for mixtures at CKA and less than $-0.03\,\mathrm{t\,ha^{-1}}$ at WG for both species. For the monoculture reference fields, the MAE is $0.179\,\mathrm{t\,ha^{-1}}$ for FB in the FB monocultures and $0.188\,\mathrm{t\,ha^{-1}}$ for SW in the SW monocultures. This is slightly higher than in the mixtures, which is expected because, in the monocultures, more of each species grows in absolute terms than in the mixtures. In return, the mixtures generally have a higher total biomass [53]. The low estimation of SW on FB monocultures between $0.001\,\mathrm{t\,ha^{-1}}$ and $0.018\,\mathrm{t\,ha^{-1}}$ and vice versa FB on SW monocultures between $0.003\,\mathrm{t\,ha^{-1}}$ and $0.017\,\mathrm{t\,ha^{-1}}$ can be considered as additional evidence that the model is able to distinguish the species with high accuracy. It can be assumed that a common weed found in both fields, *Chenopodium album*, which bears partial similarity to FB, is often incorrectly identified as FB. The mean absolute error (MAE) will be lower if there are fewer weeds or if it is included in the GEM.

In Fig. 7.2, the overall results for CKA are visualized as two scatter plots for SW and FB, where the estimations are plotted against the reference from the process-based crop growth model. The regression line is close to the optimal line with a minimal underestimation for SW (ME $= -0.026\,\mathrm{t\,ha^{-1}}$) and a minimal overestimation (ME $= 0.005\,\mathrm{t\,ha^{-1}}$) for FB. In total, the regression results are MAE $= 0.14\,\mathrm{t\,ha^{-1}}$ and $R^2 = 0.99$ for SW and MAE $= 0.10\,\mathrm{t\,ha^{-1}}$ and $R^2 = 0.98$ for FB. With this, the model is considered as accurate enough for an evaluation of generated images.

Table 7.3: Evaluation with metrics MS-SSIM, LPIPS, and FID. Each row represents a distinct IGM trained on a varying combination of conditions time (t), treatment (trt), and simulated biomass (bm); for testing, only the input image and t are varied. MS-SSIM is reported for generations with different $|\Delta t|$ filters: T_0: identity $|\Delta t| = 0$; ST: short-term $1 \leq |\Delta t| \leq 10$; LT: long-term $|\Delta t| \geq 11$. [1]Transferability check: Model trained on Mixed-CKA and applied to Mixed-WG.

	Train conds.			MS-SSIM (↑)				LPIPS (↓)	FID (↓)
	t	trt	bm	T_0	ST	LT	ø	ø	ø
Arabidopsis-S	✓	×	×	0.94	0.81	0.68	0.80	0.25	6.54
GrowliFlower	✓	×	×	0.98	0.30	0.20	0.29	0.51	20.17
Mixed-CKA	✓	×	×	0.99	0.23	0.22	0.30	0.46	20.44
Mixed-CKA	✓	✓	×	0.97	0.25	0.23	0.31	0.47	16.26
Mixed-CKA	✓	✓	✓	0.99	0.23	0.22	0.29	0.46	24.86
Mixed-WG[1]	✓	×	×	0.92	0.13	0.11	0.20	0.50	40.67

When assessing the following results, it is important to consider that they strongly rely on the accuracy of the GEMs. However, the accuracy of the GEMs is evaluated solely based on real reference images. Any discrepancy between the growth estimation of these real reference images and the data-driven predictions of the same growth stage can be attributed to two factors. First, it could be due to actual differences in plant phenotypes compared to the reference images. This is the deviation we aim to identify. Second, part of the deviation may be caused by potential small corruptions or artifacts in the artificial images, even if they pass GAN evaluation metrics. These corruptions can lead to incorrect estimations by the GEM despite the visible plant phenotypes in the artificial images being accurate. This is because the GEM was not trained on corrupted images. While it is impossible to completely avoid or quantify the second source of deviation, we strive to minimize it by augmenting the data used to train the GEM, making it more robust and less susceptible to corruption. The magnitude of the deviation can be determined for certain growth stages by comparing the biomass estimation of real images with data-driven predictions of the same growth stage as shown on the right in Fig. 7.6.

7.3.3 Time-varying image generation

The first image generation experiment will evaluate how accurately our framework predicts images of other growth stages of the plant, given an input image and a different amount of conditions used for training, as indicated in Tab. 7.3. For each prediction, conditions that match the input image are used, and a varying prediction time and the corresponding reference image are randomly picked.

Table 7.4: Plant-specific evaluation of projected leaf area (PLA) assessed by MAE and ME in the unit [% image^{-1}]. Both IGMs are trained solely on the temporal condition (t). MAE is reported for generations with different $|\Delta t|$ filters: T_0: identity $|\Delta t| = 0$; ST: short-term $1 \leq |\Delta t| \leq 10$; LT: long-term $|\Delta t| \geq 11$.

	Train conds.			MAE_{PLA}				ME_{PLA}
	t	trt	bm	T_0	ST	LT	ø	ø
Arabidopsis-S	✓	×	×	0.27	0.76	1.44	0.82	-0.32
GrowliFlower	✓	×	×	6.41	8.84	10.18	9.64	1.27

Multiple models are trained on the different datasets and with a varying combination of conditions, namely time (t), treatment (trt), and simulated biomass (bm).

Quantitative evaluation of image quality

In Tab. 7.3, the predicted image quality is evaluated using the metrics MS-SSIM, LPIPS, and FID. Across all predictions, the highest accuracies are obtained with Arabidopsis-S for all three metrics MS-SSIM = 0.8, LPIPS = 0.25, and FID = 6.54, while similarly lower overall accuracies are obtained with the GrowliFlower and MixedCrop datasets. For these, the MS-SSIM is between 0.29 and 0.31, LPIPS is between 0.46 and 0.51, and FID is between 16.26 and 24.86. Particularly remarkable is the dependence of the accuracy on the prediction distance, where MS-SSIM is higher for all datasets, the smaller $|\Delta t|$. In the case of $\Delta t = 0$, the model acts as an autoencoder, reproducing the input, also known as identity mapping. The identity mapping results show an MS-SSIM of 0.94 for Arabidopsis-S and MS-SSIM values between 0.97 and 0.99 for the Mixed-CKA models. From short-term (ST) to long-term (LT) predictions, the MS-SSIM continuously decreases to 0.20.

Plant-trait-based evaluation

Insight into the usability of predicted images can be drawn from the plant-specific evaluation results using projected leaf area (PLA) estimation for Arabidopsis-S and GrowliFlower and biomass (BM) estimation for MixedCrop.

Tab. 7.4 shows the obtained results for Arabidopsis-S and GrowliFlower in Tab. 7.4. It can be seen that MAE increases with larger $|\Delta t|$ in both cases, but the overall accuracy of $<1\,\%$ is high for Arabidopsis-S and with $<10\,\%$ slightly lower for GrowliFlower. In addition, for Arabidopsis-S, a mean error of $-0.32\,\% \approx -11\,\mathrm{mm}^2$ indicates a small mean underestimation, while GrowliFlower heads are predicted larger ME $= 1.27\,\% \approx 80\,\mathrm{cm}^2$ than the corresponding reference.

Table 7.5: Plant-specific evaluation of mixture biomasses (SW/FB) assessed by MAE and ME in the unit $\mathrm{t\,ha^{-1}}$ given for all (OA) and for mixture (Mix) fields. Each row represents a distinct IGM trained on a varying combination of conditions time (t), treatment (trt), and simulated biomass (bm); for testing, only the input image and t are varied. Overall fields, MAE is reported for generations with different $|\Delta t|$ filters: T_0: identity $|\Delta t| = 0$; ST: short-term $1 \leq |\Delta t| \leq 10$; LT: long-term $|\Delta t| \geq 11$. [1]Transferability check: Model trained on Mixed-CKA and applied to Mixed-WG.

	Train conds.				OA MAE				OA ME	Mix MAE	Mix ME
	t	trt	bm		T_0	ST	LT	ø	ø	ø	ø
Mixed-CKA	✓	×	×	SW	0.22	0.42	0.39	0.38	0.12	0.31	0.20
				FB	0.16	0.34	0.30	0.28	-0.12	0.25	-0.17
Mixed-CKA	✓	✓	×	SW	0.30	0.22	0.25	0.24	0.09	0.25	0.15
				FB	0.24	0.16	0.19	0.19	-0.13	0.24	-0.15
Mixed-CKA	✓	✓	✓	SW	0.17	0.21	0.18	0.18	-0.02	0.18	0.05
				FB	0.11	0.16	0.14	0.13	-0.01	0.15	-0.04
Mixed-WG[1]	✓	×	×	SW	0.45	1.25	1.14	1.07	0.18	1.06	0.24
				FB	0.41	0.48	0.67	0.64	-0.04	0.62	-0.11

The biomass evaluation for Mixed-CKA in Tab. 7.5 is divided into models trained with different conditions. All scores are given separately for SW and FB; moreover, average values overall plots and all mixture plots are reported. The MAE separation into different prediction distances shows that for T_0, the lowest deviations occur with a small increase to ST but a decrease (accuracy gain) for LT over ST. The overall MAE ranges from $0.13\,\mathrm{t\,ha^{-1}}$ to $0.38\,\mathrm{t\,ha^{-1}}$ and is comparable to Mix MAE, where only mixtures are considered. Thereby, overall SW MAE is always higher than FB MAE with a magnitude of up to $0.1\,\mathrm{t\,ha^{-1}}$. Noticeably, overall FB ME is negative while SW ME is positive for all models except those trained on all conditions, showing a systematic SW over- and a FB underestimation. With an increasing number of conditions, the overall MAE decreases significantly by $0.2\,\mathrm{t\,ha^{-1}}$ for SW and $0.15\,\mathrm{t\,ha^{-1}}$ for FB. Comparing the accuracy when biomass estimation is performed on predicted mixtures (last two columns of Tab. 7.5) with the accuracy when it is performed on real mixtures (first two columns of Tab. 7.2) two results are shown: First, the MAE of the predicted mixtures using the model with all conditions is slightly above the MAE of the real mixtures (SW: $+0.04\,\mathrm{t\,ha^{-1}}$, FB: $0.03\,\mathrm{t\,ha^{-1}}$). The other models trained with fewer conditions show higher deviations up to $0.17\,\mathrm{t\,ha^{-1}}$ for SW and $0.13\,\mathrm{t\,ha^{-1}}$ for FB. Second, the ME of the predicted mixtures using the model with all conditions is by a magnitude of 5 above the ME of the real mixtures.

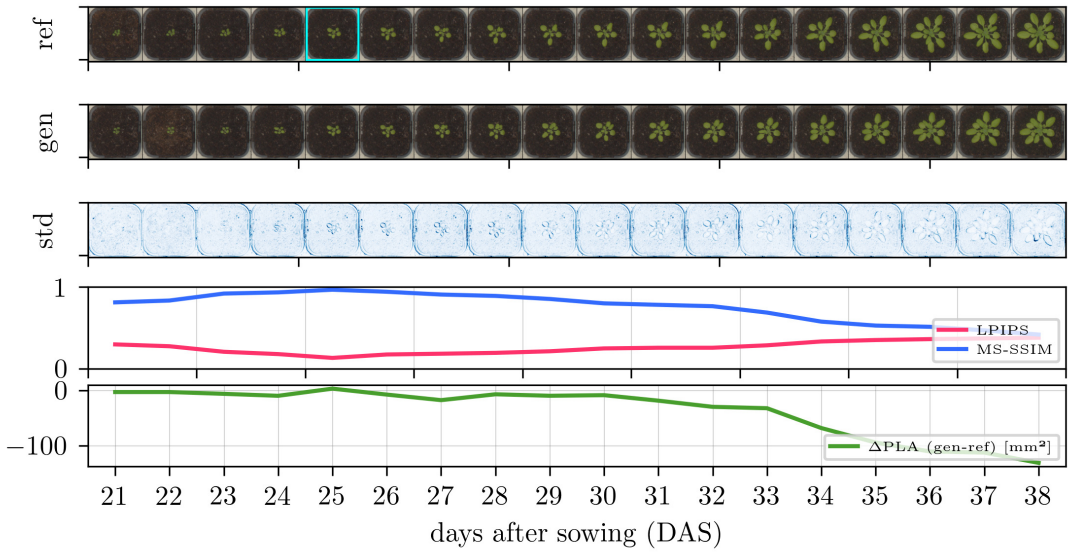


Figure 7.3: Time-varying image generation for Arabidopsis-S with, in the top row, reference images with an early growth stage as input (cyan frame), in the second row, all day-wise generated predictions, and, in the third row, standard deviation images over 10 predictions with different noise input $\boldsymbol{\epsilon}$ and otherwise constant input conditions. The two bottom rows have the quality metrics: learned perceptual image patch similarity (LPIPS), multiscale structural similarity (MS-SSIM), and the projected leaf area difference (ΔPLA).

Overall, the quantitative evaluation leads to the finding: Although the predicted images match the reference images less at large $|\Delta t|$, they represent realistic plants of their respective growth stage, as indicated by FID, and are still accurate enough to derive reasonable plant traits, as indicated by plant-specific evaluation.

Qualitative results

Further findings can be drawn from qualitative results showing selected time-varying image generation results in Fig. 7.3 for Arabidopsis-S, Fig. 7.4 for GrowliFlower, and Fig. 7.5 for Mixed-CKA, where models are used that are trained on the temporal condition only. Each figure consists of 5 rows: The first row contains a reference plant over time, where an early growth stage with a cyan frame is the input to the model in each case. The second row shows generated images by keeping except time all other conditions, including noise $\boldsymbol{\epsilon}$, constant. The third row shows the variability image, which is the standard deviation over ten predictions of the same time point with different $\boldsymbol{\epsilon}$, whereby the standard deviation is averaged over all RGB channels and overdrawn by a factor of four for clearer visualization. The darker the blue, the greater the variability for each pixel within the ten predictions. The fourth and fifth rows show each gen-ref

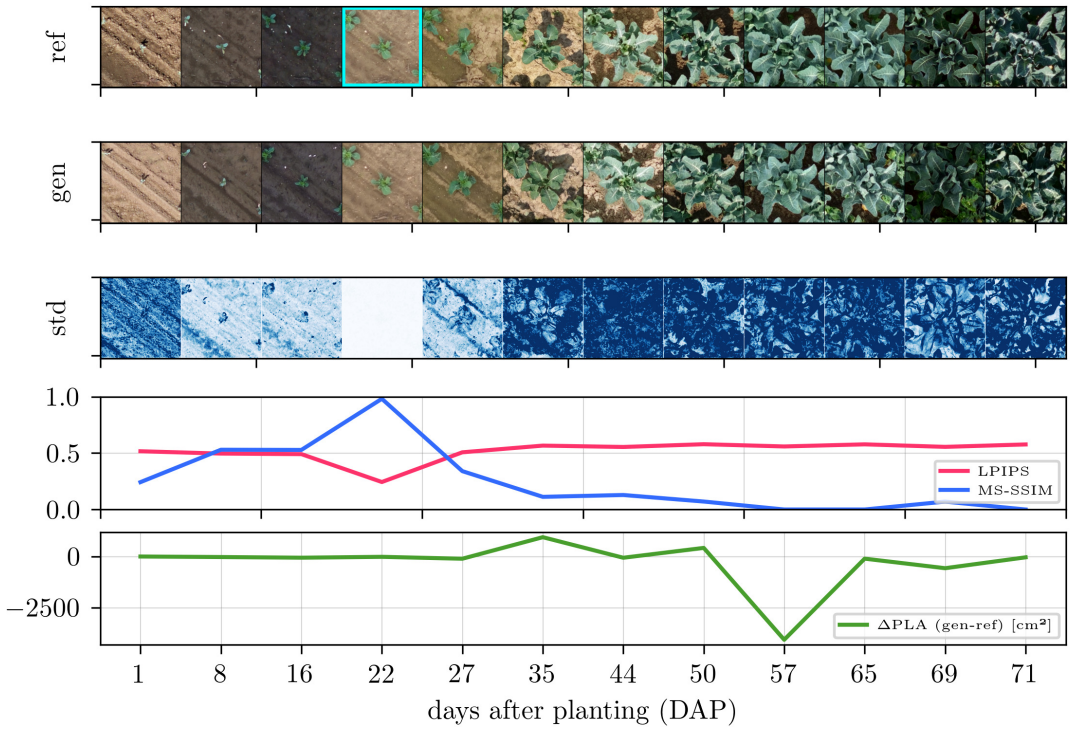


Figure 7.4: Time-varying image generation for GrowliFlower with, from top to bottom, the generated, reference, and standard deviation images, as well as the LPIPS, MS-SSIM, and ΔPLA scores. For a detailed description, see Fig. 7.3.

image pair's classical and plant-specific evaluation metrics.

For all datasets and time points, the predictions are realistic, with a few exceptions, such as the last image of GrowliFlower. In particular, the plant development is consistent over time, and a clear relation to the input image is visible. This can be determined by the orientation of the leaves (Arabidopsis-S), the position of the plants (GrowliFlower), and the field structure (Mixed-CKA). Comparing the variability images, Arabidopsis-S has the lowest pixel-wise standard deviation, followed by MixedCrop and GrowliFlower. In all cases, there is high variability at the leaf edges, where naturally the changes to plants are greatest. The LPIPS and MS-SSIM deteriorate with increasing Δt with a peak each for identity mapping. Plant property curves differ for each data set: In Arabidopsis-S, ΔPLA is close to zero until 30 DAS and then drifts into the negative range, indicating a leaf area underestimation for advanced growth stages. In GrowliFlower, the curve is close to zero with small fluctuations except for a large negative peak at 57 DAP, indicating that the leaf area could not be correctly estimated from the predicted image of this day. Similarly, for Mixed-CKA, the curves stay around zero until day 99, after which SW biomass is significantly overestimated with up to $2.5\,\mathrm{t\,ha^{-1}}$ and FB biomass is significantly underestimated with up to $-2.5\,\mathrm{t\,ha^{-1}}$. It can be concluded that, apart from some outliers, plant traits can be derived

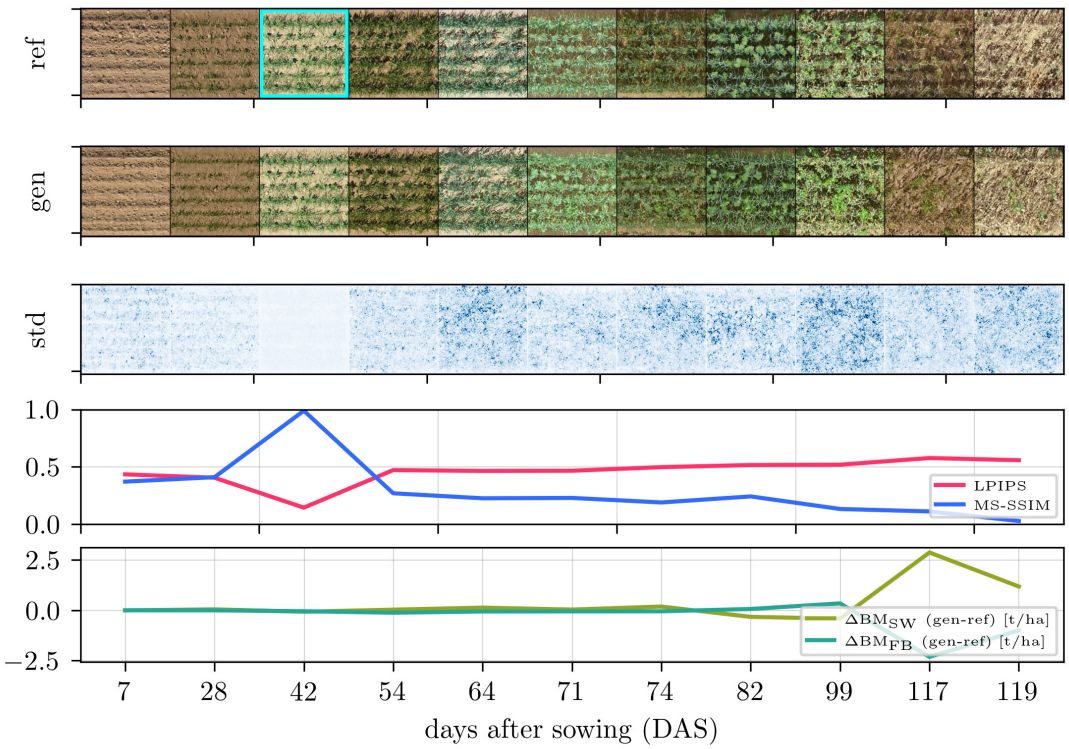


Figure 7.5: Time-varying image generation for Mixed-CKA with, from top to bottom, the generated, reference, and standard deviation images, as well as the LPIPS, MS-SSIM, and the biomass differences for spring wheat (ΔBM_{SW}) and faba bean (ΔBM_{FB}. For a detailed description, see Fig. 7.3.

with high accuracy, even from long-term predictions.

7.3.4 Comparison of process-based and data-driven model

Since there are independent reference measurements of the dried biomass ("cutting reference") for all plots at time 83 DAS for Mixed-CKA, we can compare the process-based and the data-driven CGM predictions. For both models, we use the time point 82 DAS after sowing as the prediction target, the closest image acquisition time before the biomass cuts. We select time point 28 DAS as the image input of the data-driven model because it is the first time crops are recognizable on the images (cf. Fig. 4.2). As a further input condition, we use the treatment information, which is also available to the process-based model, but not the biomass information, which is only available retrospectively. Two aspects have to be taken into account in the comparison. First, the starting conditions are not identical because the image-based model requires an input image from a previous growth stage. In contrast, the process-based model does not require an input image. Second, the models are not independent because the growth estimation part of the data-driven model was trained with the output of the

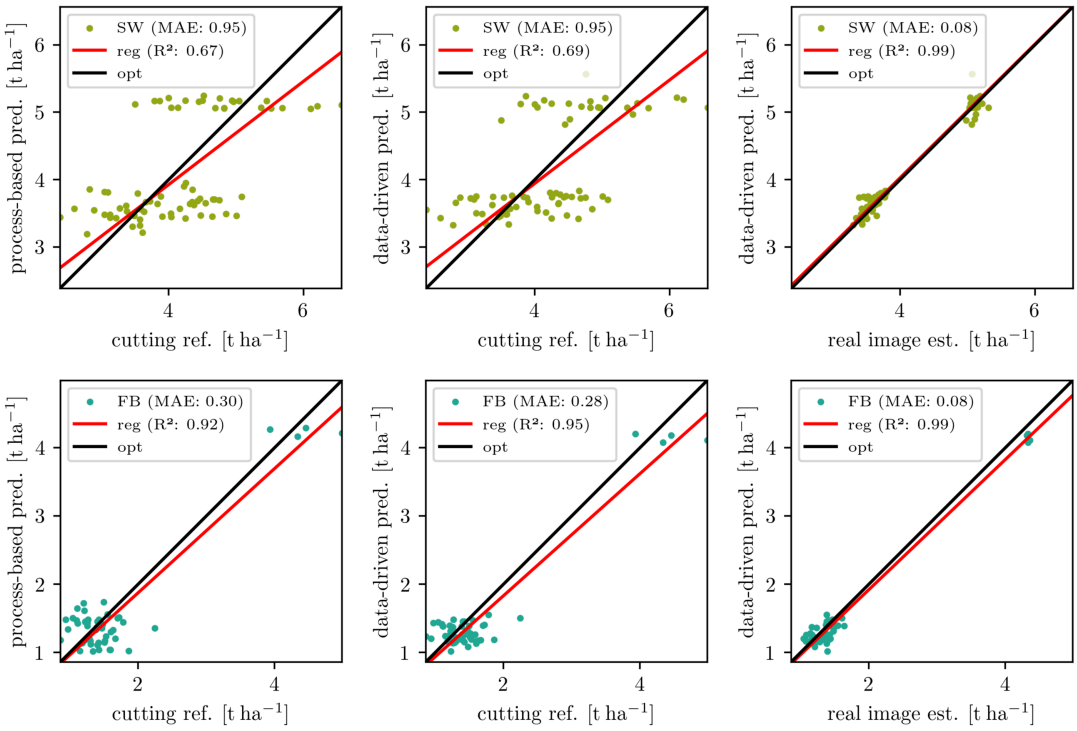


Figure 7.6: Comparison of model predictions [y-axis] for 82 DAS (left: process-based, middle+right: data-driven) with in-field biomass measurements ("cutting reference") [x-axis] at 83 DAS (left+center) and real image estimates [x-axis] at 82 DAS (right). On top are scatter plots for SW biomass, below for FB biomass.

process-based model. As a result, the data-driven model is expected to achieve, at best, the same accuracy as the process-based model when compared with the cutting reference, provided the generated images are of adequate quality. The latter is verified by comparing the estimated biomass from the data-driven prediction (generated images) with the estimated biomass from the real images from the reference day (82 DAS). If the data-driven model provides realistic predictions and the generated images are of a quality that is suitable for plant phenotyping, a high correlation can be expected.

In Fig. 7.6, the treatment-wise comparison between the process-based predictions and the cutting reference is shown on the left, between the data-driven predictions and the cutting reference in the middle, and between the data-driven predictions and the real image estimations on the right. The top row shows the SW, and the bottom row shows the FB biomasses. Two clusters can be seen in all plots: The blob with the higher biomass contains the monocultures, while the lower biomass clusters represent the mixtures. The process-based model deviates from the cutting reference for SW with MAE $= 0.95\,\mathrm{t\,ha^{-1}}$ ($R^2 = 0.67$) and for FB with MAE $= 0.30\,\mathrm{t\,ha^{-1}}$ ($R^2 = 0.92$). The pattern is similar for the data-driven model, for SW with MAE $= 0.95\,\mathrm{t\,ha^{-1}}$ ($R^2 = 0.69$) and for FB with

MAE = 0.28 t ha^{-1} ($R^2 = 0.95$). Overall, there are significantly larger MAEs for SW than for FB. In addition, the prediction range for SW is significantly narrower than the cutting reference range for both the process-based and the data-driven estimation. Focusing on the mixtures, the predicted values range between 3.2 and 4 t ha^{-1} while for the cutting reference, they range between 2.4 and 5 t ha^{-1}. This means that the actual measured variability of SW biomass between treatments is significantly larger than the predicted variability, both process-based and data-driven. Remarkably, the mean value 3.7 t ha^{-1} is identical for both models and the cutting reference. The comparison between the data-driven prediction and the estimation from the real images at time 82 DAS on the right side in Fig. 7.6 shows only a small MAE = 0.08 t ha^{-1} and a high $R^2 = 0.99$ for both SW and FB. This means that the MAE for this time point is 0.06 t ha^{-1} (SW) resp. 0.02 t ha^{-1} (FB) lower than in the comparison of all time points of the process-based model with the estimation from real images (cf. Fig. 7.2). The red regression line indicates that overall SW is slightly over- and FB slightly underestimated, which is already analyzed in Sec. 7.3.3.

7.3.5 Data-driven simulation using treatment information

The data-driven simulations on the MixedCrop dataset are intended to show the flexibility of the IGM in the presence of changing growth-influencing variables. To enable an illustrative and informative demonstration and visualization, we systematically vary the time (t) and treatment (trt) information as a condition for the Mixed-CKA dataset. We use the results to investigate and evaluate how different treatments appear in the future when something about the treatment changes starting from a certain initial condition (image). We would like to emphasize that the change in treatments performed is intended to evaluate the method and is thus limited in its realistic nature, yet aims to show that our framework applies to realistic scenarios. We expect that the estimated biomass from the data-driven simulation changes in the same direction as that of the process-based CGM, confirming the reliability of the image generations.

In particular, two simulations are conducted from the input time point of 28 DAS to 54 DAS where first, the seed density is changed from low (L) to high (H) (Fig. 7.7), and second, the faba bean cultivar is changed from Mallory (A) to Fanfare (B) (Fig. 7.8). Thus, the input image is encoded in the original treatment, but a treatment change is made to decode the simulated future plant phenotype. The figures compare the data-driven prediction without treatment change (filled bars) with the prediction including treatment change (hashed bars) and the process-based predictions for the respective target treatment (red dots). The bars represent the treatment-wise mean, and the black lines are the standard deviation. We deliberately chose an early stage as the input because the

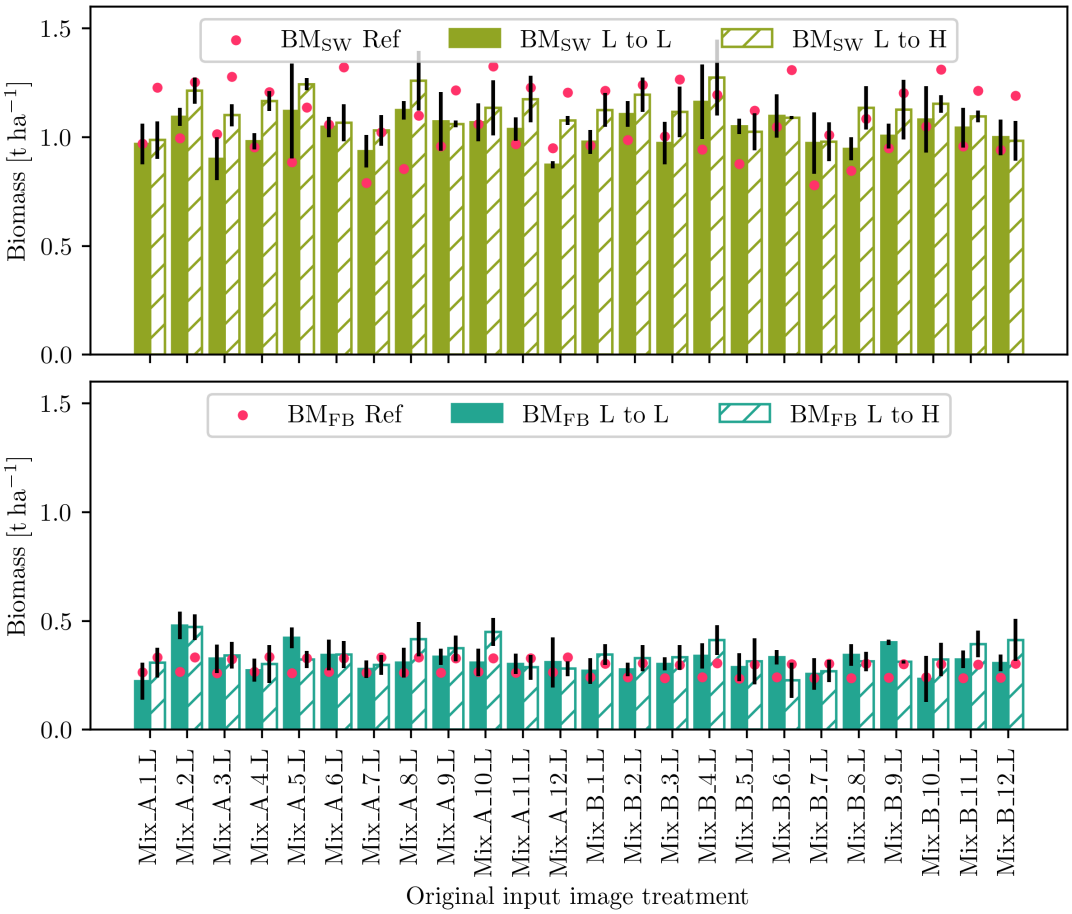


Figure 7.7: Simulating the SW (top) and FB (bottom) change from a low (L) density to a high (H) density treatment for all mixture field plots and the growth prediction step 28 DAS to 54 DAS. While filled bars represent the comparative prediction under the original treatment, hashed bars represent the simulated treatment change. Black lines symbolize the standard deviation across treatment replicates; red dots symbolize the outcome of the process-based CGM for the resp. treatments and 54 DAS.

differences in biomass between the treatments are not yet too great, and differences between the FB varieties are hardly discernible. However, we do not use DAS=7, which is bare soil, because we want to observe the spatial development of the crops. In addition, we focus on mixtures in the simulations to analyze the biomass of spring wheat and faba bean in parallel.

Focusing on the simulation of L $\rightarrow$ H in Fig. 7.7, the data-driven estimated biomass of the high-density simulated treatments (hashed bars) is higher than that of the low-density simulated ones (filled bars) for SW in 20/24 cases and for FB in 16/24 cases. The process-based biomass gain from L $\rightarrow$ H, shown by the red dots, is for SW significantly higher (0.25 t ha^{-1}) than for FB (<0.1 t ha^{-1}). Averaged across all treatments, the biomass increases for both SW and FB. Apparently, FB biomass is slightly overestimated compared to the reference in almost

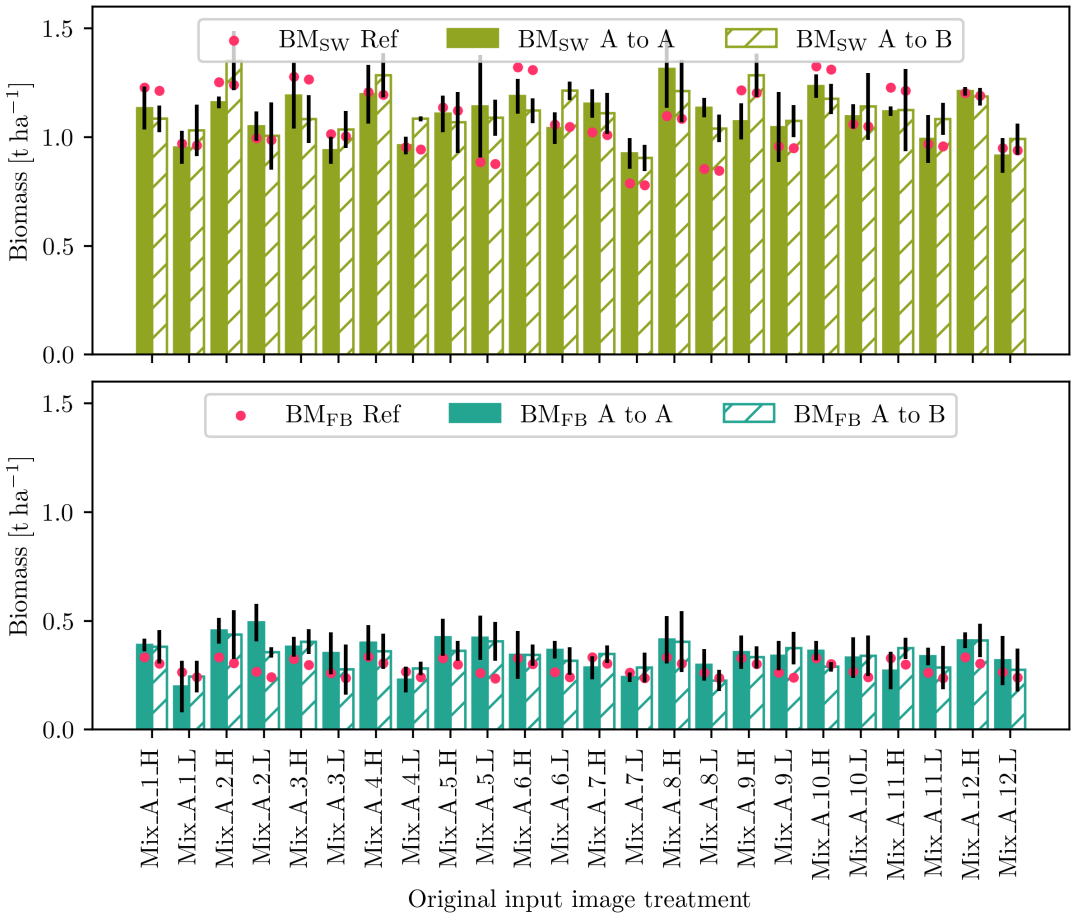


Figure 7.8: Simulating the SW (top) and FB (bottom) change from faba bean cultivar *Mallory* (A) to cultivar *Fanfare* (B) for all mixture field plots and the growth prediction step 28 DAS to 54 DAS. See Fig. 7.7 for a detailed description.

all cases, and SW biomass is often overestimated for the L → L simulation while underestimated for L → H.

The analysis of the simulation of faba bean cultivar A → B in Fig. 7.8 is more challenging because only a small loss of biomass is expected for FB and an even smaller one for SW (almost the same level), as shown by the red dots. Treatment-wise, this decrease is not visible for either SW or FB: Only slightly more than half of the treatments is the hashed bar smaller than the filled bar for both SW (13/24) and FB (15/24). In average over all treatments, the hashed bars are smaller than the filled bares, albeit in the range of the standard deviation. Comparing high- and low-density treatments, the estimated biomass from the high-density treatments is higher for SW in 10/12 cases and for FB in 7/12 cases.

Fig. 7.9 also qualitatively illustrates the structural differences in the crop rows when simulating different treatments. Besides the growth prediction step from 28 DAS to 54 DAS, two more growth prediction steps and two more treatment variations are simulated, including more unlikely scenarios, such as transforma-

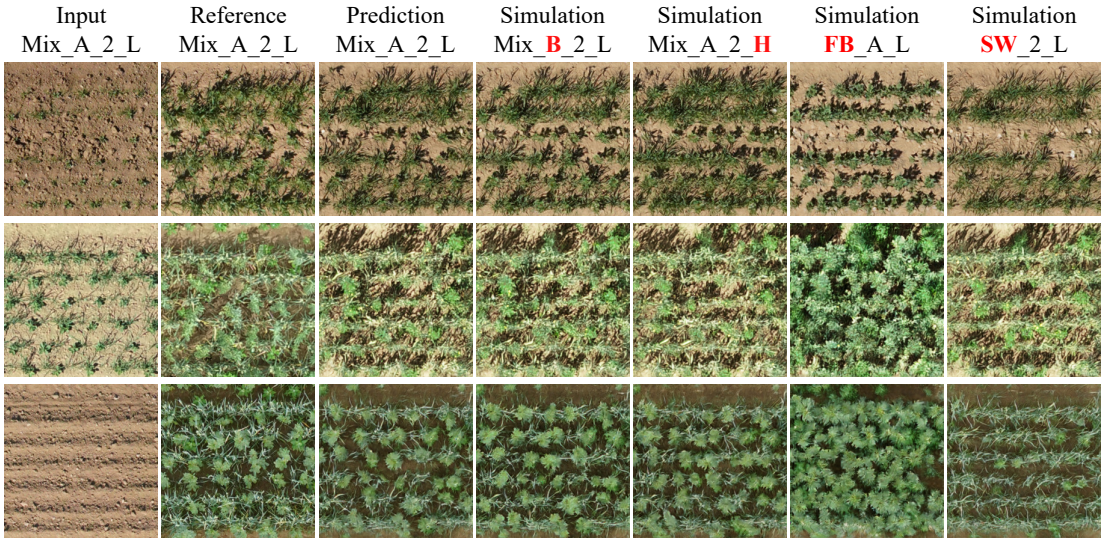


Figure 7.9: Growth simulation for different prediction steps and treatment changes in Mixed-CKA, first row 28 DAS to 45 DAS, second row 42 DAS to 99 DAS, and third row 7 DAS to 82 DAS. The first column shows the input image, the second the corresponding reference image of the future growth stage, the third the predicted image at these treatment conditions, and columns 4 to 7 show simulations of change in faba bean cultivar, density, and to monocultural reference.

tions of mixtures to monocultures. While such simulations rarely make sense from an application point of view, as long as a mixture component is not completely suppressed, it is nevertheless noteworthy to see the model visualizing such a treatment change if necessary.

7.3.6 Data-driven simulation using process-based biomass

The following biomass simulation is intended to demonstrate the capability of including dynamic output variables of a process-based CGM in our framework. For this, we use the trained Mixed-CKA model on time (t), treatment (trt), and process-based simulated biomass (bm), whereby the biomass systematically varied to get predictions for different possible SW and FB biomass ratios. The time is randomly varied, so the simulation is performed over all growth stages by choosing a random prediction time point for each input mixture image and re-adjusting its biomass ratio. The starting point for the simulation is the biomasses calculated dynamically from the process-based CGM for each time point and treatment, $\mathrm{BM_{SW}} = \mathrm{BM_{FB}} = 100$ %. While the IGM was trained with a fixed biomass value attached to each reference image, we will demonstrate that almost any combination of biomass ratios can be chosen for inference as long as they are within the range of the training data.

Fig. 7.10 shows MAE and ME respectively for SW and FB and different simulated biomass ratios, where the original composition (100:100) is shown in the

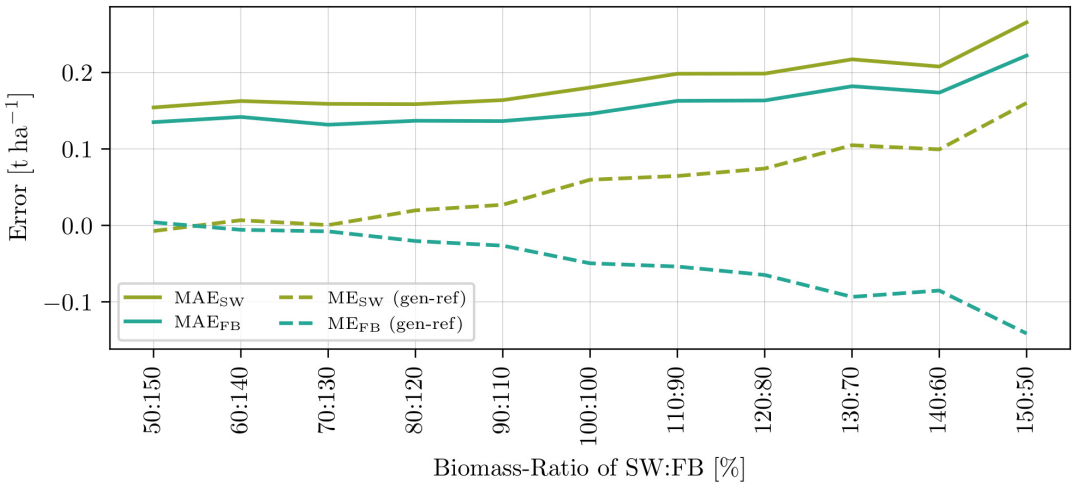


Figure 7.10: Comparing MAE and ME for Mixed-CKA image generations from 28 DAS to 54 DAS with different simulated spring wheat (SW) to faba bean (FB) biomass ratios.

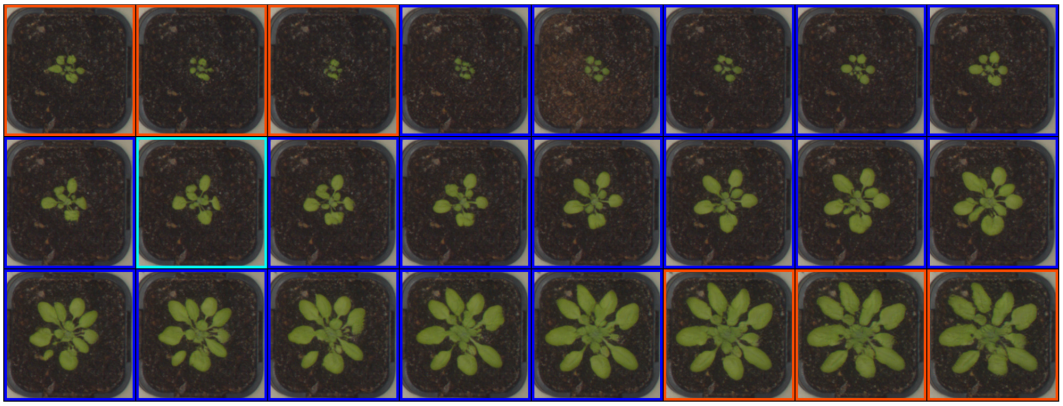


Figure 7.11: Daily Arabidopsis-S predictions from 18 DAS to 41 DAS including temporal OOD images. The input image has a cyan frame, the in-distribution images a blue frame, and the OOD images an orange frame.

middle, to the left, BM_{FB} increases and to the right BM_{SW}. This is accordingly also noticeable in the ME: If the BM fraction for SW and FB increases, more biomass is also estimated in the predicted image, and the ME increases. So ME_{SW} rises to the right, and the ME_{FB} rises to the left. This means that the SW biomass must be reduced and the FB biomass increased in the input of this IGM, compared to what the process-based model has simulated in order to achieve a minimum prediction error.

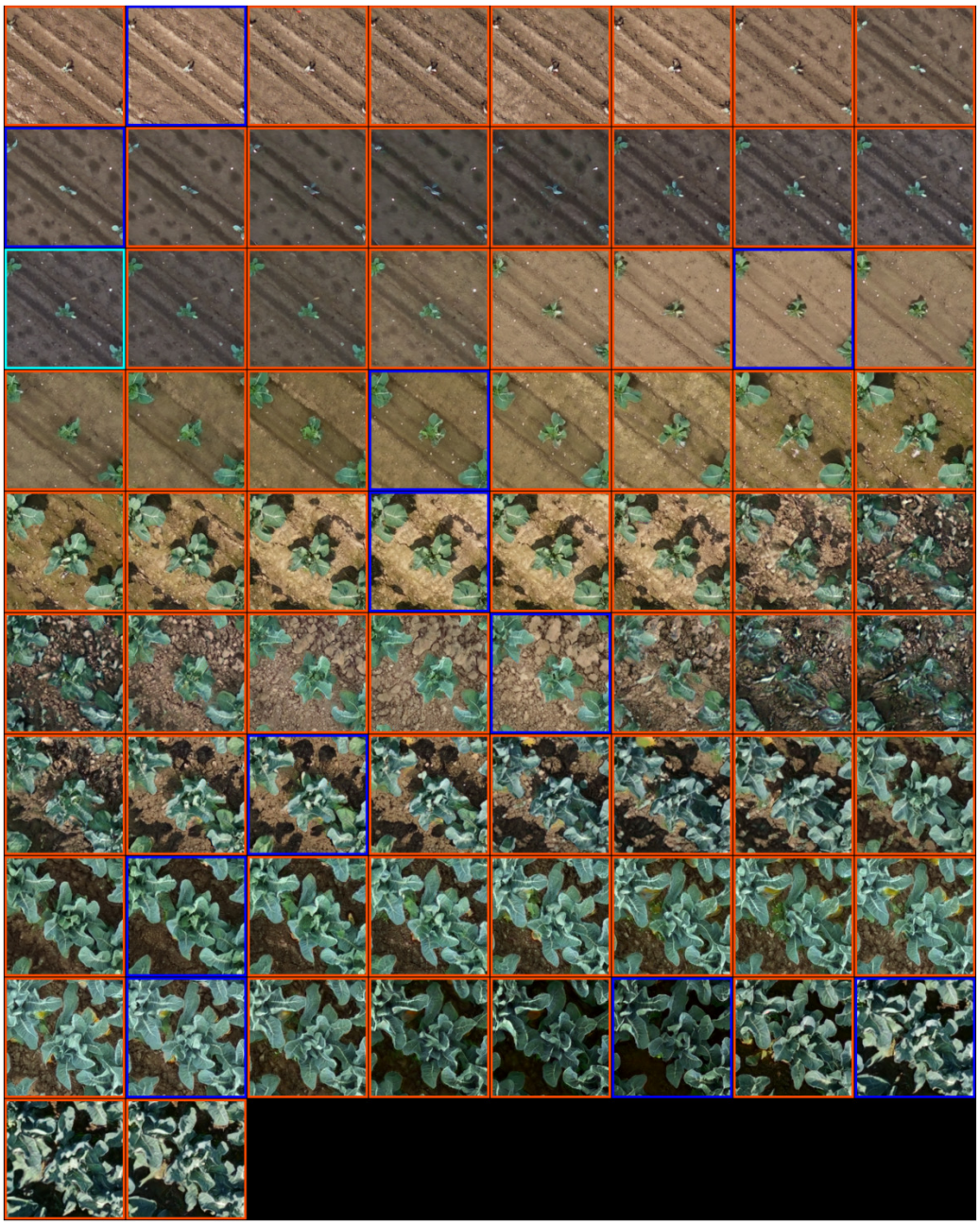


Figure 7.12: Daily GrowliFlower predictions from 0 DAP to 73 DAP including temporal OOD images. The input image has a cyan frame, the in-distribution images a blue frame, and the OOD images an orange frame.

7.3.7 Spatial and temporal out-of-distribution generations

Temporal out-of-distribution

By temporal out-of-distribution (OOD) images, visualized in Fig. 7.11, Fig. 7.12, and Fig. 7.13, we refer to images of growth stages that do not exist in the training

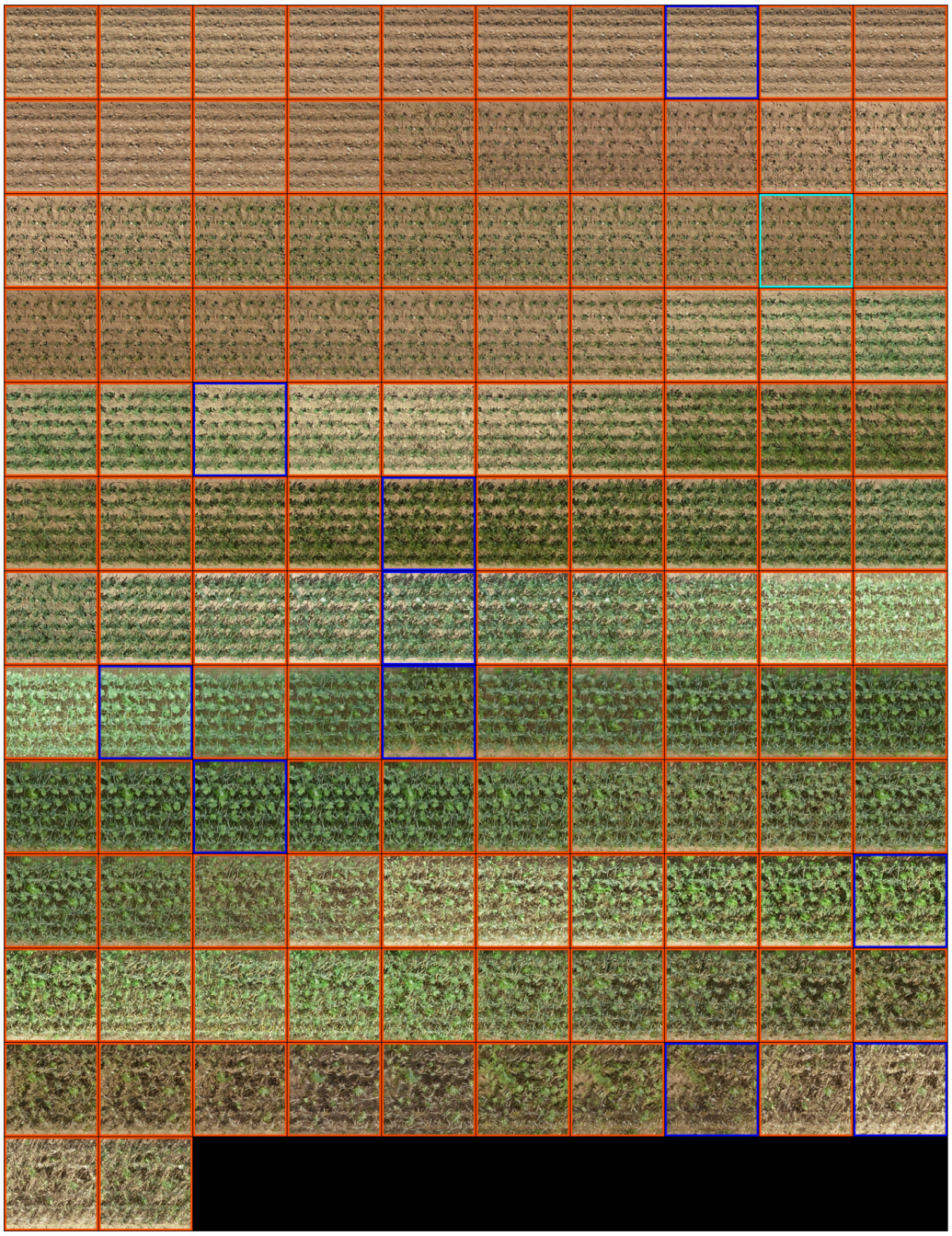


Figure 7.13: Daily Mixed-CKA predictions from 0 DAS to 121 DAS including temporal OOD images. The input image has a cyan frame, the in-distribution images a blue frame, and the OOD images an orange frame.

dataset. This needs to be distinguished from in-distribution images, whose growth stages exist in the training dataset. We use the models from the respective dataset trained solely on input image and time as conditions and keep the input image and the noise constant for the visualizations from the entire growth period. So

we iterate over time and generate interpolations if the newly generated image lies between two training images and extrapolations if it lies temporally off the training period (early and late growth stages). The time increases by one day per image from top left to bottom right. The input image has a cyan frame, the in-distribution images a blue frame, and the OOD images an orange frame. While challenging to evaluate quantitatively because no reference images are available, the consistency of the time series in terms of continuous growth development can be examined.

Overall, the interpolation shows a continuous and reasonable growth trend. However, there are exceptions: In GrowliFlower, in the out-of-distribution images in rows 5 and 6 of Fig. 7.12 on the right. Here, plants almost vanish in front of a darker background and then become larger again. A similar is visible in Mixed-CKA, where the canopy increases in row 3 of Fig. 7.13 and decreases towards the input image. These growth curves are not realistic and could be caused by leaving the data manifold at these growth stages. Remarkably, the out-of-distribution images of GrowliFlower and Mixed-CKA show a smooth transition between in-distribution images in terms of brightness and contrast, which makes the newly generated time series look reasonable.

For extrapolations, most predictions are also realistic since plants continue to grow in the short-term extrapolated future and shrink when going back in time. However, there are exceptions. For example, the early two growth stages of Arabidopsis-S get larger with decreasing age, which is not realistic. Since the observation for GrowliFlower does not begin with sowing but with the planting of seedlings (0 DAP) and the time cannot be negative (there is no positional encoding for $t_{\text{gen}}<0$), it is not possible to extrapolate further into the past. So, no images of bare soil can be generated.

Spatial out-of-distribution: Transferability to new site

With a transferability experiment on the MixedCrop experiment, we aim to investigate the accuracy drop with which the model trained for Mixed-CKA, which takes time (t) as input condition, can be applied to the Mixed-WG site. The basic requirements are given by the same image size, resolution, crop species, and treatments (see Sec. 4.2.4). However, this attempt to transfer the growth behavior of Mixed-CKA to images of Mixed-WG poses three main challenges. First, the growth behavior of conventionally managed CKA differs substantially from that of organically managed WG, as indicated, for instance, by weed abundance. Second, the spectral image properties are completely different for each time point, so both sites have their own "style". Third, images were not taken simultaneously during the growing season at both locations, resulting in images from Mixed-WG being spatially and temporally out-of-distribution (OOD).

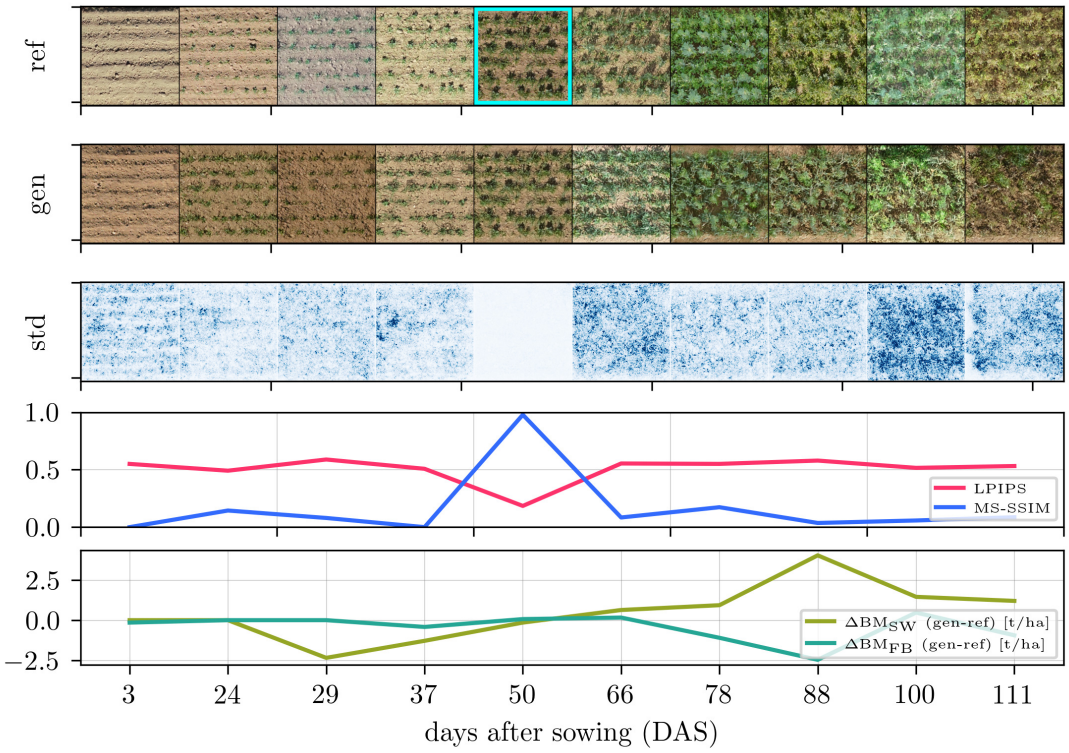


Figure 7.14: Transferability with prediction results for Mixed-WG input image 50 DAS lying spectrally less far away from the 54 DAS-images of the training distribution (Mixed-CKA). The predicted images are qualitatively appealing, but they do not compare well with the reference because the crops of Mixed-CKA and Mixed-WG have different growth patterns.

Tab. 7.3 and Tab. 7.5 show the transferability quality measured by all evaluation metrics in the bottom line each. It can be seen that the results show significantly lower accuracies than the ones produced by models trained and tested on Mixed-CKA. However, the identity predictions still show a high MS-SSIM of 0.92. Transferability fails when the spectral differences between the test image and the nearby time points in the training dataset are too large, such as 29 DAS of Mixed-WG, as Fig. 7.15 illustrates. Some of the predicted images become blurry, and holes appear in the crop rows, which also causes the biomass estimation to give unreliable, non-usable results. Likewise, Fig. 7.14 demonstrates that the model can produce reasonable results despite spatio-temporal OOD, where, compared to Fig. 7.15, the same field patch but a different input image (21 days later) is used.

The reason for the less accurate results lies in the first two aforementioned challenges, which led to the predicted images not being quantitatively comparable to the reference images. Since the model only knows the style of CKA, but the reference images are in the style of WG, better scores were not expected. Focusing more on qualitative results, the third challenge of temporal OOD leads to corrupted results when the input image is significantly different from the style

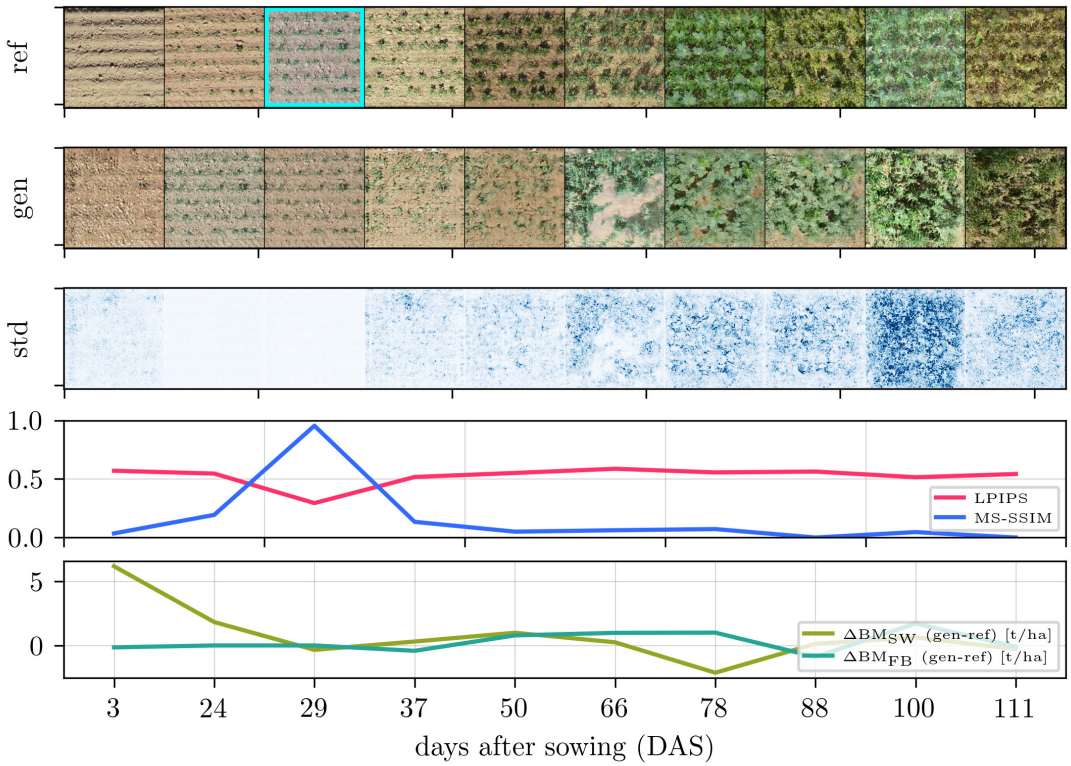


Figure 7.15: Transferability fails with predictions for Mixed-WG caused by input image 29 DAS lying spectrally too far out of the training distribution (Mixed-CKA images).

of the temporally nearest CKA image but is otherwise reliable. It shows both failed predictions and reasonable transfer examples, first for time points for which reference images are available, even if they do not match the reference, and second for the entire growing period.

7.4 Discussion

7.4.1 Analysis of image generations

Quantitative discussion

In image generation, where only the time is varied, but growth influencing factors of the output match the input, there are considerable variations in the accuracies between the data sets. It is noticeable that Arabidopsis-S has better values in all metrics except T_0 than GrowliFlower and Mixed-CKA, which can be attributed to the daily recording times and controlled laboratory conditions with constant light and no weather effects. The identity mapping (T_0) is worse than the other datasets because, in Arabidopsis-S, multiple images were taken per day, which means it is not a strict identity mapping. However, this can be altered by

changing the model time unit from days to hours. The MS-SSIM decrease from T_0 over ST to LT means the less far the model predicts into the future or past, the better the predicted images match the reference. Particularly, an MS-SSIM below 0.3 implies less similarity between predicted and reference images. In parallel, the FID for all models, including ST and LT predictions, is below 25, which can be considered good image quality. This is expected because, with increasing prediction steps, detailed plant phenotype appearances, like leaf counts and orientations, are increasingly difficult to predict. In contrast, general structural traits, like plant positions and overall sizes, can be predicted more accurately.

When analyzing the biomass from the generated MixedCrop images, significant differences between the SW and FB components emerge, for which there are two explanations. First, SW has a generally higher MAE magnitude than FB, which can be attributed to the overall higher SW biomass level in the field. Second, there is a systematic overestimation of SW and an underestimation of FB indicated by the ME. We assume this is due to the unbalanced dataset: there are significantly more SW than FB monocultures. Apparently, the IGM copes worse with this unbalanced dataset than the GEM, as FB plants are structurally more complex and, therefore, more readily quantifiable but more difficult to generate.

Besides, MAE and ME decrease significantly for both SW and FB as more conditions are added to the model. This can be explained by the model being better informed about the crop growth behavior if it receives more growth-influencing factors and can thus become more accurate. There is a loss of accuracy from identity mapping to short-term predictions but no significant loss from short-term to long-term predictions. Thus, long-term predictions can be considered valuable for phenotyping applications.

Qualitative discussion

Two important insights emerge from the qualitative analysis. First, a strong consistency of the generated images over time is given, which is visible in Arabidopsis-S and GrowliFlower through leaf orientations but also through neighboring plants and in Mixed-CKA through certain crop patterns such as small gaps (Fig. 7.5: second crop row, right) or weeds (third and fourth crop row, center). Second, the dependence of the generated images on the input is visible for all datasets, particularly in the position of the plants and crop rows and by granules on the ground, which can be found on the input image as well as on several generated images.

While the variability images show realistic variability at the leaf edges, they also reveal a limitation in the image generation: While the identity mapping has no or extremely low variability, as expected, no continuous increase in variability over time is evident, leading to overconfidence at large Δt where variability would

be expected to be significantly higher.

The parallel examination of MS-SSIM and LPIPS with the images confirms the findings from the quantitative results: Despite the images becoming less consistent with the reference as the prediction distance increases, there is neither a general decrease in visual quality nor a general decrease in the accuracy of the estimated plant traits for time-varying predictions.

7.4.2 Comparison of image generation results with TransGrow

Compared to the TransGrow model introduced in Chap. 6, image generation in this chapter is not based on an input sequence but on a single point in time. Remarkably, a consistent growth trend can still be observed when images with otherwise constant growth influencing factors are generated over time. In addition, the identity mapping is significantly improved, as evidenced by the low pixel-wise standard deviation in the variability images. In return, more distant time points drift more strongly from the reference, while with TransGrow, the entire time series can be kept close to the reference over the entire growth period by inserting additional input images.

Comparing the image quality using average MS-SSIM and FID (Tab. 7.3) with TransGrow when generating with $I_{\text{in}} = 1$ (Tab. 6.2) for the real field datasets GrowliFlower and Mixed-CKA (for Arabidopsis-S no comparability is given due to the special shuffling augmentation) substantial differences can be identified. The MS-SSIM (GrowliFlower: +0.06, Mixed-CKA: +0.08) and the FID are both drastically improved (GrowliFlower: -235.89, Mixed-CKA: -71.13). This is because TransGrow incorporates a combined CNN-transformer encoder, while this chapter's multi-modal image generation framework only has a CNN encoder. The additional transformer encoder provides flexibility in the input, but it accounts for over 10 million additional model parameters, which increases the demand for the dataset size.

Please note: Although only in-distribution generations were shown for TransGrow, growth stages not present in the training dataset (out-of-distribution) can also be requested. Due to the methodically identical representation of time by sinusoidal positional encoding, experiments have provided similar results.

7.4.3 Data-driven and process-based comparison

Some findings can be taken away from the comparisons of data-driven and process-based approaches: Mainly, process-based and data-driven models achieve similar accuracy despite the long-term prediction 54 days into the future. Both models can quantify differences between mixtures and monocultures of the same growth

stage but are hardly sensitive to differences between the mixture treatments. They only achieve the prediction of a correct mean value, which can be explained by the fact that many cultivar differences occur randomly. While seed density and environment have a large influence on the absolute mixture effect (AME) and the land equivalent ratio (LER), the influences of SW cultivar and FB cultivar are not significant [153].

In general, a machine learning model (growth estimation from images) can hardly be better than the training data (process-based output), which accounts for the similar pattern in the left and middle scatter plots of Fig. 7.6. If there were other biomass reference data for each time point, we could use it to train the GEM and become completely independent of the process-based model. It is conceivable that such biomass reference data might be available in the future and outperform the process-based model as it is trained with measurements instead of simulations. However, these biomass reference data would need to be available in advance and ideally be highly diverse to allow generalization across different environments.

7.4.4 Analysis of image simulations

Treatment simulations

Two treatment simulations were conducted: first, the simulation of density change L $\rightarrow$ H, and second, the simulation of faba bean cultivar change A $\rightarrow$ B. Both simulation results demonstrate that even small changes in the growth-influencing factors affect the predicted images. Thereby, the reliability of the simulations is supported by the overall biomass increase from L $\rightarrow$ H treatments and decrease from faba bean cultivar A $\rightarrow$ B. If in Fig. 7.7 and Fig. 7.8, the prediction change (filled to hashed bar) for individual treatments does not correspond to the expected change (red dots), there are three possible interpretations. First, although the treatment condition is considered in the IGM, its influence might not be strong enough, so the differences in the generated images are not sufficiently prominent. Second, the density resp. cultivar appearance of the input image might already be too prominent, making it difficult to change the growth stage later; e.g., plants cannot arise from anywhere. Third, the differences between low and high-density treatments, respectively, between faba bean cultivars A and B, are less clear in reality than the dynamic CGM suggests. In fact, the FB biomass gain for L $\rightarrow$ H and the FB/SW biomass loss for A $\rightarrow$ B is below the accuracy level of the biomass estimation (cf. Tab. 7.5), which can explain why a clear trend in biomass changes is not particularly apparent for these cases. Apart from these specific experiments, we see the potential to simulate further treatment changes or their effects, e.g., weed cover. This varies over the growing season and can

be estimated quickly in categorical measures (low, medium, high), allowing crop growth predictions adapted to current field conditions.

Biomass simulations

The biomass simulations show that this condition reasonably affects the output image: A higher SW simulated biomass in the framework's input leads to a higher SW prediction in the output and for FB accordingly. So, the predictions realistically depend on the input conditions.

It also demonstrates the capability of our framework to generate images that plausibly explain the output of a process-based model. The minimum MAE/ME is not reached at 100:100, mainly due to the slight dataset bias towards SW and the resulting under-prediction of FB plants in the images, as already discussed. Assuming an unbiased IGM, this type of analysis can serve to improve the calibration of the process-based model and bring it closer to image-based field observations: If the minimum MAE deviates from the expected minimum (in this case, 100:100), the process-based CGM could be adjusted in this direction or, in other words, complemented by the knowledge gained from the data-driven model. Note that other dynamic growth-influencing variables, like climatic conditions, can be used instead of process-based time-varying biomass, which could lead to even more feasible simulations.

7.4.5 Generalizability assessment

The difficulties of generalization are that (1) different external management conditions are present, (2) spectral image properties, i.e., the data set style varies, and (3) the training time points do not match the inference time points. Inference data are, therefore, in many respects, out-of-distribution of the training data.

Assuming the availability of this data, it would be reasonable to add management factors and styles as auxiliary conditions via conditional batch normalization in the IGM. More generally, domain knowledge in the form of site-dependent context variables could be included that influence style and plant growth itself [21]. Different climatic conditions in general and weather in particular are also considered to be site-dependent context variables. While this requires a larger training dataset spanning multiple sites, it will ensure even better transferability and help to merge multiple plant time series affected by various factors influencing factors into a more generic data-driven crop growth model.

7.5 Conclusion

In this chapter, we have shown the capabilities of multi-conditional growth simulation using three datasets: Arabidopsis-S, GrowliFlower, and MixedCrop. For this purpose, in the first step, we combined several conditions of different types (discrete, continuous, categorical) in an IGM, which is a conditional Wasserstein generative adversarial network (CWGAN), to generate multiple realistic, high-quality images over time based on a single input image. In the second step of growth estimation, we showed that along with classical GAN image evaluation metrics, plant-specific traits such as projected leaf area or biomass can be derived from the generated images and used for evaluation. The results for MixedCrop were compared with a dynamic process-based crop growth model. Here, the combination of data-driven crop growth models, which strongly incorporate the spatio-temporal above-ground phenotype changes, and a process-based crop growth model, which considers the theoretical plant growth knowledge, leads to a better understanding of the crop mixture dynamics. Quantitative and qualitative simulations provide a comprehensive tool to investigate how various treatments influence the above-ground phenotype of crop mixtures and their dry matter.

In particular, the integration of process-based model output into a data-driven CGM is useful for making crop growth predictions more accessible or even for recalibrating process-based models. The experiments show that the dried biomass can be estimated more accurately from predicted images the more growth influencing factors are considered, such as in our case, the field treatment or process-based simulated biomasses. Incorporating all available conditions into the IGM enables accurate estimation of plant traits in predicted (artificial) images, comparable to the accuracy achieved with real images.

Although the additional variability images show the largest variability at the leaf edges, which is realistic, we see space for improvement in the stochasticity integration for long-term growth predictions. Since predictions far in the future lead to significant overconfidence in the IGM, the weighting of the stochastic and deterministic model input should be adaptively controlled depending on the growth prediction step. In addition, the challenge of large spectral differences within an image sequence and between sites ("dataset styles") should be addressed for better model generalizability.

Chapter 8

Conclusion

Crop Growth Models (CGMs) play a crucial role in the transformation towards sustainable agriculture. They help to gain a better understanding of the development of crops and enable growth predictions. This not only fundamentally increases planning certainty but also facilitates the optimization of cultivation and thus minimizes environmental impacts. For example, by using resources such as water, fertilizers, and pesticides as needed and at the right times during the growing season, or by identifying risks such as pest infestation at an early stage. They also play an important role in research into modern cultivation systems such as crop mixtures: By using simulation to find constellations of plant species that complement each other well, environmental impacts can be minimized, and yields increased simultaneously. In view of global warming and increasing extreme weather conditions, they make a significant contribution to finding plants that adapt well to changing climatic conditions and thus increase agriculture's resilience.

While process-based growth models were mainly used in the past, in which the relationships were defined using expert knowledge, data-driven models are playing an increasingly important role nowadays. This development is driven in particular by the availability of data from smart agricultural machines and drones, which can be processed using modern machine-learning methods. The main focus of this work was on the processing of image data, both in the input and output of the developed data-driven models that represent Conditional Generative Adversarial Network (CGAN).

In a two-step process, we first generate artificial images of future plant growth stages based on one or more images of an early time point and other growth-influencing factors, and, in the second step, we perform plant phenotyping, i.e., relevant plant traits are derived from the images. Compared to traditional CGMs, where target parameters are estimated directly, the artificially generated images provide an essential added value in three ways. First, they serve as artificial

sensor data and can thus be used for a wide variety of phenotyping applications. Second, they visualize the spatial plant distribution, which is especially relevant for crop mixtures and an essential prerequisite for targeted in-field interventions. Third, they increase the reliability of complex CGMs and, thus, the explainability of a CGM.

8.1 Summary of key contributions

The basic prerequisite for the developed CGMs is the data (Chap. 4) in this work, in particular paired and sequential RGB images linked to other growth-influencing variables. We have presented several datasets of different complexity on which the experiments in this work were performed, namely image data of *Arabidopsis thaliana*, *Brassica oleracea var. botrytis* (cauliflower) and crop mixtures, which consist of *Triticum aestivum* (spring wheat) and *Vicia faba* (faba bean). We have compared the data and analyzed which basic requirements must be met for image data to be suitable for crop growth modeling, focusing specifically on image resolution and perspective, measurement setup, lighting conditions, and image alignment.

In Chap. 5, we first used paired datasets to show that an image-to-image translation-based CGM, which solely uses images as a condition, can perform realistic and reasonable long-term predictions of a fixed growth step. The input and target images are structurally significantly different compared to state-of-the-art work and the same model can be used for predictions of different growth stages. We have shown that the CGMs can generate realistic predictions for different treatments without explicitly including the treatment information as a condition in the model. This is possible even with very early growth stages in the input, where the effects of treatment differences are still minimal and barely noticeable based on plant traits. In addition to classic GAN evaluation metrics, we also evaluated the generated images using projected leaf area, which was derived in the generated images and reference images using instance segmentation, thus demonstrating that the images are suitable as artificial sensor data.

The focus of Chap. 6 was to increase the flexibility of the CGM by processing multiple input images simultaneously and generating images of arbitrary growth stages. In contrast to related works that work with regular input sequences, the presented model TransGrow can process non-equidistant input sequences of different lengths that regularly occur in observing agricultural fields. For this purpose, we present a combination of convolutional and transformer layers for spatio-temporal encoding of the input sequence. The use of global positional encoding is essential because it allows any target image to be generated. This way, inter- and extrapolations are possible, such as generating a potential future

growth stage. By using a Wasserstein Generative Adversarial Network (WGAN) instead of a classical GAN for image generation, we have shown that several weaknesses, such as mode-collapse and low output diversity, can be mitigated by not suppressing the stochastic model component.

In Chap. 7, we addressed the problem of multi-modal conditioning by using conditional batch normalization to integrate additional growth-influencing factors of different data types (discrete, continuous, categorical) into the CGM in addition to the input image. We show that the more conditions are integrated into the model, i.e., the better the plant growth is described with additional factors, the lower the deviation of the generated image from the reference image. By recombining the conditions for inference in constellations that do not exist in the training conditions, simulations could be conducted. Such simulations are essential for crop mixture research, as they help us better understand which varieties should be mixed in which treatments to achieve a positive mixing effect. Finally, we displayed that a process-based model's output can be integrated into the data-driven CGM. The combination of both models can provide higher spatial specificity and can indicate the need for re-calibration of the process-based model.

In summary, we have addressed the key challenges of data-driven crop growth modeling in this work. Within a two-step image generation and growth estimation process, the contributions are mainly in the image generation part. There, we have shown that we can generate realistic time-differentiated images that, although structurally significantly different from input images, still inherit essential information such as plant size, leaf orientation, and plant positions from the input. It was demonstrated that non-equidistant sequences in the input could be processed, and a consistent output time series could be generated. Within multi-modal conditioning, the linking of image data with other growth-influencing factors and process-based output has been achieved, enabling crop growth simulations. Because crop growth is not deterministic, the generation of an output distribution and associated variability maps is an important contribution enabled by the use of WGAN. Finally, in the second part of CGM, we performed a plant-trait-based evaluation by carrying out dataset-specific phenotyping on the generated images, thus demonstrating that the generated images are suitable as artificial sensor data.

8.2 Open source contributions

Two open-source frameworks have been released, featuring data-driven CGMs based on sequential input (Chap. 6) and multi-modal conditions (Chap. 7). Additionally, two RGB image datasets of crop mixtures, collected in 2020 as part

of the PhenoRob project and displaying pre-processed field patches, are available open-source on the PhenoRoam platform.

- **TransGrow**, Python, presented in Chap. 6
  `https://github.com/luked12/transgrow`
- **CGANs for Crop Growth Simulations**, Python, presented in Chap. 7
  `https://github.com/luked12/crop-growth-cgan`
- **Mixed-CKA**, Sequential RGB image dataset, introduced in Chap. 4
  `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/751c10c4-b6dc-4bcc-bc8c-c0fc5920887a`
- **Mixed-WG**, Sequential RGB image dataset, introduced in Chap. 4
  `https://phenoroam.phenorob.de/geonetwork/srv/eng/catalog.search#/metadata/d9d0434f-7864-435e-9c75-56102d9332cb`

8.3 Future Work

8.3.1 Diffusion models

While this work methodically focuses on different GANs variants, denoising diffusion probabilistic models [32] have recently drawn attention because they deliver promising results in the field of image generation. In many cases, e.g., with large datasets such as ImageNet, they achieve better FID scores than state-of-the-art GAN approaches [154].

A denoising diffusion probabilistic model works by progressively adding and then removing noise to generate realistic data samples. The process involves two main phases: In the forward diffusion process, starting from real data, the model sequentially adds Gaussian noise over a series of time steps, transforming the data into a pure noise distribution. Each step is controlled by a fixed noise schedule, resulting in a series of increasingly noisy versions of the original data. For the reverse denoising process, a neural network is trained to reverse the noise addition step-by-step. Given a noisy sample at a certain time step, the model predicts a less noisy sample from the previous time step. By iterating this denoising process from the final noisy state back to the original data distribution, the model generates new, realistic data samples. During training, the model learns the parameters of the reverse process by minimizing the difference between the predicted denoised samples and the actual samples. This ensures that the generated data closely matches the real data distribution.

A more efficient variant of the described method, namely a latent diffusion model [155], we have implemented for the MixedCrop dataset. With latent diffusion models, the diffusion process does not take place in the image space but

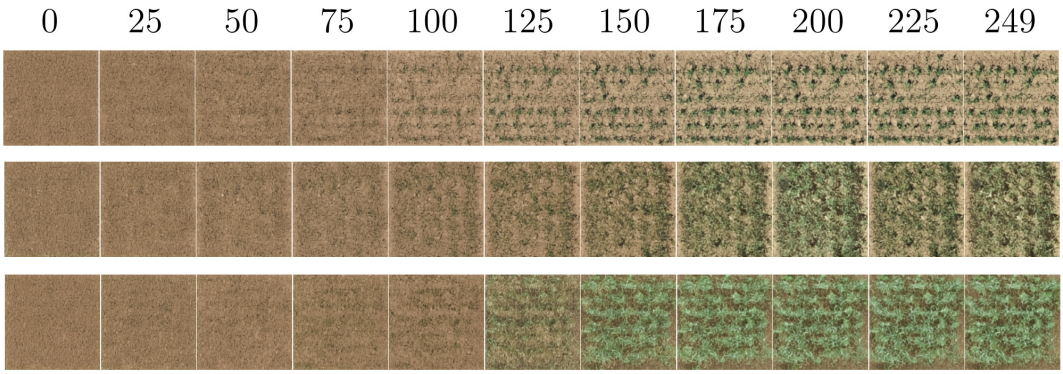


Figure 8.1: Latent diffusion generations over a noise schedule with 250 steps for MixedCrop starting with complete noise (step 0) up to no noise (step 249). The latent diffusion model is conditioned on the growth stage (top: 42 DAS, middle: 54 DAS, bottom: 82 DAS) and the treatment (top: FB, middle: SW, bottom: Mix).

in the latent space of an AE that is trained separately. Vector Quantized AE, which generate a discrete latent space, have proven particularly useful [156].

In Fig. 8.1, we have visualized three examples of denoising over 250 steps, whereby the growth stage and the treatment are provided. Both are embedded (using Φ_t and Φ_c as in Eq. 7.3), added to the noise time point embedding, and then inserted into the denoising neural network as a condition. Although there is no image of an earlier growth stage included in this case, inserting images as a condition is generally feasible [155].

Overall, latent diffusion models are very training-intensive compared to GANs but have the advantage that the training is very stable and intuitive due to the optimization of usually only one MSE loss. In addition, they can effectively model complex data distributions [32], which can potentially address the problem of too low output variability in long-term predictions as discussed in Sec. 7.4.

8.3.2 Hybrid models

There are many different types of CGMs and, in particular, several ways to combine expert/domain knowledge from Process-Based crop growth Models (PBMs) with information obtained from big (remote sensing) data in Data-Driven crop growth Models (DDMs), as described in Sec. 3.2. In this work, we have created a serial interface in Chap. 7 by further processing the output of a PBM in the input of the DDM. Further benefits could be drawn from even deeper levels of integration, in particular from combined modular CGMs. Here, certain PBM modules can be replaced by DDM, especially those that represent empirical processes or benefit significantly from information based on remote sensing data, e.g., light interception modules, the prediction of yield responses to several climate factors, or the linking of spatial image data with PBM intermediate results [73]. Conversely,

modules that can be described well on a process basis should not be adopted by DDMs so that no additional uncertainty is introduced into the model and the interpretability of the whole CGM is not affected [73].

Another approach to introducing DDMs modules while maintaining the greatest possible consistency with existing process-based contexts is physics-informed machine learning [157]. For example, a neural network might be trained to predict a system's behavior while constrained by physical equations. This can significantly help to incorporate remote sensing data and thereby improve model robustness and generalizability, while current CGMs are often limited to specific crop types, climates, and study areas [158].

8.3.3 Further perspectives on data-driven modeling

We intend to address further aspects crucial for the generalizability of data-driven image generation for crop growth modeling and its usefulness in agricultural practice. Since data is central to DDMs, the points mainly revolve around the scope, type, and curation of datasets.

CGMs become particularly valuable for agricultural practice when they can be transferred to different environments, i.e., are generalizable. Such generalizability experiments are presented in this work by applying CGMs developed at one site to another site with image data that are spatially and temporally outside the training distribution. Also, the growth simulations, where conditions are combined that do not occur in the training dataset demonstrate generalizability. Ultimately, however, the growth behaviors mapped in the CGMs are limited to the area of the training data, which is a general challenge of ML models [158].

To tackle this problem, the amount of image data from real environments, including the associated growth-influencing factors, must be increased. This means a higher spatial distribution and a temporal distribution, i.e., several different growth periods. Besides the size, it is important to maintain diversity, consistency, and unbiasedness [159]. Otherwise, there is no possibility of developing image-generating CGMs depending on changing climatic conditions, for which there is considerable demand [1].

One way to increase the amount of data is to integrate additional data sources, such as satellite images or hand-held images, depending on the required resolution. Farmers can capture such images conveniently with mobile devices. However, meeting the high data requirements is challenging, especially for images from hand-held devices (see Sec. 4.1). Likewise, with multiple data sources, the calibration and alignment of the imaging sensors among each other must be ensured. If this is the case, multi-spectral images can also be taken into account because they provide added value in detecting plant diseases [93].

During the research period of this work, we have experienced related research

projects that produce high-quality datasets that are not prepared and processed in a way that makes them accessible and reusable. Therefore, we see a great need for the development of research data management platforms such as FAIRagro [160] for agrosystems research, which define high-quality criteria of data. This is in line with the idea of data-centric machine learning, in which the focus is not on model optimization but on optimizing data creation and curation to improve the problem definition in the first place [159].

Bibliography

[1] G. Fischer, "World food and agriculture to 2030/50," in *Proc. of the FAO Expert Meeting on How to Feed the World in 2050*, 2009, pp. 24–26.

[2] R. Gebbers and V. I. Adamchuk, "Precision agriculture and food security," *Science*, vol. 327, no. 5967, pp. 828–831, 2010. DOI: 10.1126/science.1183899.

[3] A. C. Tyagi, "Towards a second green revolution," *Irrigation and Drainage*, vol. 65, no. 4, pp. 388–389, 2016. DOI: 10.1002/ird.2076.

[4] T. Duckett, S. Pearson, S. Blackmore, B. Grieve, W.-H. Chen, G. Cielniak, J. Cleaversmith, J. Dai, S. Davis, C. Fox, *et al.*, "Agricultural robotics: The future of robotic agriculture," *arXiv preprint arXiv:1806.06762*, 2018. DOI: 10.48550/arXiv.1806.06762.

[5] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018. DOI: 10.1016/j.compag.2018.02.016.

[6] N. Zhu, X. Liu, Z. Liu, K. Hu, Y. Wang, J. Tan, M. Huang, Q. Zhu, X. Ji, Y. Jiang, *et al.*, "Deep learning for smart agriculture: Concepts, tools, applications, and opportunities," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 32–44, 2018. DOI: 10.25165/j.ijabe.20181103.4475.

[7] S. A. Tsaftaris, M. Minervini, and H. Scharr, "Machine learning for plant phenotyping needs image processing," *Trends in plant science*, vol. 21, no. 12, pp. 989–991, 2016. DOI: 10.1016/j.tplants.2016.10.002.

[8] R. Pieruschka and U. Schurr, "Plant phenotyping: Past, present, and future," *Plant Phenomics*, 2019. DOI: 10.34133/2019/7507131.

[9] A. Burkart, V. Hecht, T. Kraska, and U. Rascher, "Phenological analysis of unmanned aerial vehicle based time series of barley imagery with high temporal resolution," *Precision Agriculture*, vol. 19, no. 1, pp. 134–146, 2018. DOI: 10.1007/s11119-017-9504-y.

[10] S. Chang, U. Lee, M. J. Hong, Y. D. Jo, and J.-B. Kim, "Time-series growth prediction model based on u-net and machine learning in arabidopsis," *Frontiers in Plant Science*, vol. 12, 2021, ISSN: 1664-462X. DOI: 10.3389/fpls.2021.721512.

[11] A. Chlingaryan, S. Sukkarieh, and B. Whelan, "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review," *Computers and Electronics in Agriculture*, vol. 151, pp. 61–69, 2018. DOI: 10.1016/j.compag.2018.05.012.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.

[13] R. Roscher, B. Bohn, M. Duarte, and J. Garcke, "Explain it to me - facing remote sensing challenges in the bio- and geosciences with explainable machine learning," in *Proc. of the ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-3-2020, 2020, pp. 817–824. DOI: 10.5194/isprs-annals-V-3-2020-817-2020.

[14] L. Drees, L. V. Junker-Frohn, J. Kierdorf, and R. Roscher, "Temporal prediction and evaluation of brassica growth in the field using conditional generative adversarial networks," *Computers and Electronics in Agriculture*, vol. 190, p. 106 415, 2021, ISSN: 0168-1699. DOI: 10.1016/j.compag.2021.106415.

[15] L. Drees, I. Weber, M. Rußwurm, and R. Roscher, "Time dependent image generation of plants from incomplete sequences with cnn-transformer," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, and I. Ihrke, Eds., Cham: Springer International Publishing, 2022, pp. 495–510, ISBN: 978-3-031-16788-1. DOI: 10.1007/978-3-031-16788-1_30.

[16] L. Drees, D. T. Demie, M. R. Paul, J. Leonhardt, S. J. Seidel, T. F. Döring, and R. Roscher, "Data-driven crop growth simulation on time-varying generated images using multi-conditional generative adversarial networks," *Plant Methods*, vol. 20, no. 1, p. 93, Jun. 2024, ISSN: 1746-4811. DOI: 10.1186/s13007-024-01205-3.

[17] J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, and R. Roscher, "Behind the leaves: Estimation of occluded grapevine berries with conditional generative adversarial networks," *Frontiers in Artificial Intelligence*, vol. 5, 2022. DOI: 10.3389/frai.2022.830026.

[18] J. Leonhardt, L. Drees, P. Jung, and R. Roscher, "Probabilistic biomass estimation with conditional generative adversarial networks," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, and I. Ihrke, Eds., Cham: Springer International Publishing, 2022, pp. 479–494, ISBN: 978-3-031-16788-1. DOI: https://doi.org/10.1007/978-3-031-16788-1_29.

[19] Q. Marashdeh, L. Drees, and R. Roscher, "Semantic uav image segmentation of mixed cropping fields," in *Proc. of the Dreiländertagung der DGPF, der OVG und der SGPF in Dreseden - Publikationen der DGPF*, vol. 30, 2022, pp. 140–148.

[20] M. Miranda, L. Drees, and R. Roscher, "Controlled multi-modal image generation for plant growth modeling," in *Proc. of the IEEE International Conference on Pattern Recognition (ICPR)*, 2022, pp. 5118–5124. DOI: 10.1109/ICPR56361.2022.9956115.

[21] J. Leonhardt, L. Drees, J. Gall, and R. Roscher, "Leveraging bioclimatic context for supervised and self-supervised land cover classification," in *Proc. of the DAGM German Conference on Pattern Recognition (GCPR)*, 2023. DOI: 10.1007/978-3-031-54605-1_15.

[22] R. Roscher, L. Drees, and S. Wenzel, "Sparse representation-based archetypal graphs for spectral clustering," in *Proc. of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2017, pp. 2203–2206. DOI: 10.1109/IGARSS.2017.8127425.

[23] L. Drees, R. Roscher, and S. Wenzel, "Archetypal analysis for sparse representation-based hyperspectral sub-pixel quantification," *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 5, pp. 279–286, 2018. DOI: 10.14358/PERS.84.5.279.

[24] L. Drees, J. Kusche, and R. Roscher, "Multi-modal deep learning with sentinel-3 observations for the detection of oceanic internal waves," in *Proc. of the ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-2-2020, 2020, pp. 813–820. DOI: 10.5194/isprs-annals-V-2-2020-813-2020.

[25] R. Roscher, M. Volpi, C. Mallet, L. Drees, and J. D. Wegner, "Semcity toulouse: A benchmark for building instance segmentation in satellite images," *Proc. of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, pp. 109–116, 2020. DOI: 10.5194/isprs-annals-V-5-2020-109-2020.

[26] J. Kierdorf, T. T. Stomberg, L. Drees, U. Rascher, and R. Roscher, "Investigating the contribution of image time series observations to cauliflower harvest-readiness prediction," *Frontiers in Artificial Intelligence*, vol. 7, Sep. 2024, ISSN: 2624-8212. DOI: 10.3389/frai.2024.1416323.

[27] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 14, 2001.

[28] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, "Autoregressive image generation using residual quantization," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 523–11 532. DOI: 10.1109/CVPR52688.2022.01123.

[29] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," *Proc. of the Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. DOI: 10.48550/arXiv.1312.6114.

[31] E. G. Tabak and E. Vanden-Eijnden, "Density estimation by dual ascent of the log-likelihood," *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 217–233, 2010. DOI: 10.4310/CMS.2010.v8.n1.a11.

[32] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851. DOI: 10.48550/arXiv.2006.11239.

[33] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991. DOI: 10.1002/aic.690370209.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017. DOI: 10.48550/arXiv.1701.07875.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 5769–5779, ISBN: 9781510860964.

[36] A. Borji, "Pros and cons of gan evaluation measures: New developments," *Computer Vision and Image Understanding*, vol. 215, p. 103 329, 2022. DOI: 10.1016/j.cviu.2021.103329.

[37] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068.

[38] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6626–6637.

[39] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. of the Asilomar Conference on Signals, Systems & Computers*, IEEE, vol. 2, 2003, pp. 1398–1402. DOI: 10.1109/ACSSC.2003.1292216.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. DOI: 10.48550/arXiv.1409.1556.

[41] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 2234–2242.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969. DOI: 10.1109/ICCV.2017.322.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[45] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (gans) for image augmentation in agriculture: A systematic review," *Computers and Electronics in Agriculture*, vol. 200, p. 107 208, 2022. DOI: 10.1016/j.compag.2022.107208.

[46] J. Sihag and D. Prakash, "A review: Importance of various modeling techniques in agriculture/crop production," in *Proc. of the Soft Computing: Theories and Applications (SoCTA) 2017*, vol. 742, Springer Singapore, 2019, pp. 699–707. DOI: 10.1007/978-981-13-0589-4_66.

[47] F. Shah and W. Wu, "Soil and crop management strategies to ensure higher crop productivity within sustainable environments," *Sustainability*, vol. 11, no. 5, p. 1485, 2019. DOI: 10.3390/su11051485.

[48] A. Di Paola, R. Valentini, and M. Santini, "An overview of available crop growth and yield models for studies and assessments in agriculture," *Journal of the Science of Food and Agriculture*, vol. 96, no. 3, pp. 709–714, 2016. DOI: 10.1002/jsfa.7359.

[49] D. Wurr, J. R. Fellows, and R. Hiron, "The influence of field environmental conditions on the growth and development of four cauliflower cultivars," *Journal of Horticultural Science*, vol. 65, no. 5, pp. 565–572, 1990. DOI: 10.1080/00221589.1990.11516094.

[50] T. Wheeler, R. Ellis, P. Hadley, and J. Morison, "Effects of co2, temperature and their interaction on the growth, development and yield of cauliflower (brassica oleracea l. botrytis)," *Scientia Horticulturae*, vol. 60, no. 3-4, pp. 181–197, 1995. DOI: 10.1016/0304-4238(94)00725-U.

[51] J. E. Olesen and K. Grevsen, "A simulation model of climate effects on plant productivity and variability in cauliflower (Brassica oleracea L. botrytis)," *Scientia Horticulturae*, vol. 83, no. 2, pp. 83–107, 2000. DOI: 10.1016/S0304-4238(99)00068-0.

[52] J. Kierdorf, L. V. Junker-Frohn, M. Delaney, M. D. Olave, A. Burkart, H. Jaenicke, O. Muller, U. Rascher, and R. Roscher, "Growliflower: An image time-series dataset for growth analysis of cauliflower," *Journal of Field Robotics*, vol. 40, no. 2, pp. 173–192, 2023. DOI: 10.1002/rob.22122.

[53] M. R. Paul, D. T. Demie, S. J. Seidel, and T. F. Döring, "Effects of spring wheat/faba bean mixtures on early crop development," *Plant and Soil*, pp. 1–16, 2023. DOI: 10.1007/s11104-023-06111-6.

[54] E. S. Jensen, M. B. Peoples, and H. Hauggaard-Nielsen, "Faba bean in cropping systems," *Field Crops Research*, vol. 115, no. 3, pp. 203–216, 2010, Faba Beans in Sustainable Agriculture, ISSN: 0378-4290. DOI: 10.1016/j.fcr.2009.10.008.

[55] L. Li, S.-M. Li, J.-H. Sun, L.-L. Zhou, X.-G. Bao, H.-G. Zhang, and F.-S. Zhang, "Diversity enhances agricultural productivity via rhizosphere phosphorus facilitation on phosphorus-deficient soils," *Proceedings of the National Academy of Sciences*, vol. 104, no. 27, pp. 11 192–11 196, 2007. DOI: 10.1073/pnas.0704591104.

[56] M. Peoples, J. Brockwell, D. Herridge, I. Rochester, B. Alves, S. Urquiaga, R. Boddey, F. Dakora, S. Bhattarai, S. Maskey, *et al.*, "The contributions of nitrogen-fixing crop legumes to the productivity of agricultural systems," *Symbiosis*, vol. 48, pp. 1–17, 2009. DOI: 10.1007/BF03179980.

[57] L. Bedoussac, E.-P. Journet, H. Hauggaard-Nielsen, C. Naudin, G. Corre-Hellou, E. S. Jensen, L. Prieur, and E. Justes, "Ecological principles underlying the increase of productivity achieved by cereal-grain legume intercrops in organic farming. a review," *Agronomy for sustainable development*, vol. 35, pp. 911–935, 2015. DOI: 10.1007/s13593-014-0277-7.

[58] Y. Yu, T.-J. Stomph, D. Makowski, L. Zhang, and W. Van Der Werf, "A meta-analysis of relative crop yields in cereal/legume mixtures suggests options for management," *Field Crops Research*, vol. 198, pp. 269–279, 2016. DOI: 10.1016/j.fcr.2016.08.001.

[59] D. Sarkar, S. K. Kar, A. Chattopadhyay, A. Rakshit, V. K. Tripathi, P. K. Dubey, P. C. Abhilash, *et al.*, "Low input sustainable agriculture: A viable climate-smart option for boosting food production in a warming world," *Ecological Indicators*, vol. 115, p. 106 412, 2020. DOI: 10.1016/j.ecolind.2020.106412.

[60] P. Pandey, V. Irulappan, M. V. Bagavathiannan, and M. Senthil-Kumar, "Impact of combined abiotic and biotic stresses on plant growth and avenues for crop improvement by exploiting physio-morphological traits," *Frontiers in Plant Science*, vol. 8, p. 537, 2017. DOI: 10.3389/fpls.2017.00537.

[61] H. Kage, H. Stützel, and C. Alt, "Predicting dry matter production of cauliflower (Brassica oleracea L. botrytis) under unstressed conditions: Part II. comparison of light use efficiency and photosynthesis–respiration based modules," *Scientia horticulturae*, vol. 87, no. 3, pp. 171–190, 2001. DOI: 10.1016/S0304-4238(00)00180-1.

[62] P. Miller, W. Lanier, and S. Brandt, "Using growing degree days to predict plant stages," *Montana State University (MT200103 AG 7/2001)*, vol. 59717, no. 406, pp. 994–2721, 2001.

[63] B. Maestrini, G. Mimić, P. A. van Oort, K. Jindo, S. Brdar, I. N. Athanasiadis, and F. K. van Evert, "Mixing process-based and data-driven approaches in yield prediction," *European Journal of Agronomy*, vol. 139, p. 126 569, 2022. DOI: 10.1016/j.eja.2022.126569.

[64] S. J. Seidel, T. Palosuo, P. Thorburn, and D. Wallach, "Towards improved calibration of crop models–where are we now and where should we go?" *European Journal of Agronomy*, vol. 94, pp. 25–35, 2018. DOI: 10.1016/j.eja.2018.01.006.

[65] E. Vanuytrecht, D. Raes, P. Steduto, T. C. Hsiao, E. Fereres, L. K. Heng, M. G. Vila, and P. M. Moreno, "Aquacrop: Fao's crop water productivity and yield response model," *Environmental Modelling & Software*, vol. 62, pp. 351–360, 2014. DOI: 10.1016/j.envsoft.2014.08.005.

[66] B. A. Keating, P. S. Carberry, G. L. Hammer, M. E. Probert, M. J. Robertson, D. Holzworth, N. I. Huth, J. N. Hargreaves, H. Meinke, Z. Hochman, *et al.*, "An overview of apsim, a model designed for farming systems simulation," *European journal of agronomy*, vol. 18, no. 3-4, pp. 267–288, 2003. DOI: 10.1016/S1161-0301(02)00108-9.

[67] J. W. Jones, G. Hoogenboom, C. H. Porter, K. J. Boote, W. D. Batchelor, L. Hunt, P. W. Wilkens, U. Singh, A. J. Gijsman, and J. T. Ritchie, "The dssat cropping system model," *European journal of agronomy*, vol. 18, no. 3-4, pp. 235–265, 2003.

[68] K. J. Boote, J. W. Jones, and G. Hoogenboom, "Simulation of crop growth: Cropgro model," in *Agricultural systems modeling and simulation*, CRC Press, 2018, pp. 651–692. DOI: 10.1201/9781482269765-18.

[69] A. Übelhör, S. Munz, S. Graeff-Hönninger, and W. Claupein, "Evaluation of the cropgro model for white cabbage production under temperate european climate conditions," *Scientia Horticulturae*, vol. 182, pp. 110–118, 2015. DOI: 10.1016/j.scienta.2014.11.019.

[70] A. Enders, M. Vianna, T. Gaiser, G. Krauss, H. Webber, A. K. Srivastava, S. J. Seidel, A. Tewes, E. E. Rezaei, and F. Ewert, "Simplace—a versatile modelling and simulation framework for sustainable crops and agroecosystems," *in silico Plants*, vol. 5, no. 1, pp. 1–18, 2023. DOI: 10.1093/insilicoplants/diad006.

[71] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42 200–42 216, 2020. DOI: 10.1109/ACCESS.2020.2976199.

[72] T. Van Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning: A systematic literature review," *Computers and Electronics in Agriculture*, vol. 177, p. 105 709, 2020. DOI: 10.1016/j.compag.2020.105709.

[73] N. Zhang, X. Zhou, M. Kang, B.-G. Hu, E. Heuvelink, and L. F. Marcelis, "Machine learning versus crop growth models: An ally, not a rival," *AoB Plants*, vol. 15, no. 2, plac061, 2023. DOI: 10.1093/aobpla/plac061.

[74] J. Huang, J. L. Gómez-Dans, H. Huang, H. Ma, Q. Wu, P. E. Lewis, S. Liang, Z. Chen, J.-H. Xue, Y. Wu, *et al.*, "Assimilation of remote sensing into crop growth models: Current status and perspectives," *Agricultural and forest meteorology*, vol. 276, p. 107 609, 2019. DOI: 10.1016/j.agrformet.2019.06.008.

[75] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Transactions on knowledge and data engineering*, vol. 29, no. 10, pp. 2318–2331, 2017. DOI: 10.1109/TKDE.2017.2720168.

[76] P. Feng, B. Wang, D. Li Liu, C. Waters, D. Xiao, L. Shi, and Q. Yu, "Dynamic wheat yield forecasts are improved by a hybrid approach using a biophysical model and machine learning technique," *Agricultural and Forest Meteorology*, vol. 285, p. 107 922, 2020. DOI: 10.1016/j.agrformet.2020.107922.

[77] M. Shahhosseini, G. Hu, I. Huber, and S. V. Archontoulis, "Coupling machine learning and crop modeling improves crop yield prediction in the us corn belt," *Scientific reports*, vol. 11, no. 1, p. 1606, 2021. DOI: 10.1038/s41598-020-80820-1.

[78] K. Johansen, M. Morton, Y. Malbeteau, *et al.*, "Predicting biomass and yield at harvest of salt-stressed tomato plants using uav imagery," in *Proc. of the ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, Copernicus GmbH, 2019, pp. 407–411. DOI: 10.5194/isprs-archives-XLII-2-W13-407-2019.

[79] P. Nevavuori, N. Narra, and T. Lipping, "Crop yield prediction with deep convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 163, p. 104 859, 2019, ISSN: 0168-1699. DOI: 10.1016/j.compag.2019.104859.

[80] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0.

[81] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017. DOI: 10.48550/arXiv.1706.03762.

[83] J. Bendig, K. Yu, H. Aasen, A. Bolten, S. Bennertz, J. Broscheit, M. L. Gnyp, and G. Bareth, "Combining uav-based plant height from crop surface models, visible, and near infrared vegetation indices for biomass monitoring in barley," *International Journal of Applied Earth Observation and Geoinformation*, vol. 39, pp. 79–87, 2015. DOI: 10.1016/j.jag.2015.02.012.

[84] M. Watt, F. Fiorani, B. Usadel, U. Rascher, O. Muller, and U. Schurr, "Phenotyping: New windows into the plant for breeders," *Annual Review of Plant Biology*, vol. 71, 2020. DOI: 10.1146/annurev-arplant-042916-041124.

[85] R. Barth, J. Hemming, and E. J. van Henten, "Improved part segmentation performance by optimising realism of synthetic images using cycle generative adversarial networks," *arXiv preprint arXiv:1803.06301*, 2018. DOI: 10.48550/arXiv.1803.06301.

[86] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, and H. T. Campus, "Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants.," in *Proc. of the British Machine Vision Conference (BMVC)*, 2018, p. 324.

[87] S. L. Madsen, M. Dyrmann, R. N. Jørgensen, and H. Karstoft, "Generating artificial images of plant seedlings using generative adversarial networks," *Biosystems Engineering*, vol. 187, pp. 147–159, 2019. DOI: 10.1016/j.biosystemseng.2019.09.005.

[88] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Computers and Electronics in Agriculture*, vol. 168, pp. 105–117, 2020. DOI: 10.1016/j.compag.2019.105117.

[89] D. Gogoll, P. Lottes, J. Weyler, N. Petrinic, and C. Stachniss, "Unsupervised domain adaptation for transferring plant classification systems to new field environments, crops, and robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2636–2642. DOI: 10.1109/IROS45743.2020.9341277.

[90] K. Yamamoto, T. Togami, and N. Yamaguchi, "Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture," *Sensors*, vol. 17, no. 11, p. 2557, 2017. DOI: 10.3390/s17112557.

[91] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proc. of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 63–79. DOI: 10.1007/978-3-030-11021-5_5.

[92] Q. H. Cap, H. Tani, S. Kagiwada, H. Uga, and H. Iyatomi, "Lassr: Effective super-resolution method for plant disease diagnosis," *Computers and Electronics in Agriculture*, vol. 187, p. 106 271, 2021. DOI: 10.1016/j.compag.2021.106271.

[93] A. Foerster, J. Behley, J. Behmann, and R. Roscher, "Hyperspectral plant disease forecasting using generative adversarial networks," in *Proc. of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019, pp. 1793–1796. DOI: 10.1109/IGARSS.2019.8898749.

[94] T. Hamamoto, H. Uchiyama, A. Shimada, and R.-i. Taniguchi, "Rgb-d images based 3d plant growth prediction by sequential images-to-images translation with plant priors," in *Proc. of the International Joint Conference on Computer Vision, Imaging and Computer Graphics*, Springer, 2020, pp. 334–352. DOI: 10.1007/978-3-030-94893-1_15.

[95] T. Kim, S.-H. Lee, and J.-O. Kim, "A novel shape based plant growth prediction algorithm using deep learning and spatial transformation," *IEEE Access*, vol. 10, pp. 37 731–37 742, 2022. DOI: 10.1109/ACCESS.2022.3165211.

[96] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[97] R. Yasrab, J. Zhang, P. Smyth, and M. P. Pound, "Predicting plant growth from time-series data using deep learning," *Remote Sensing*, vol. 13, no. 3, p. 331, 2021. DOI: 10.3390/rs13030331.

[98] S. Aigner and M. Körner, "Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans," in *Proc. of the ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W16, 2019, pp. 3–11. DOI: 10.5194/isprs-archives-XLII-2-W16-3-2019.

[99] J. Bell and H. M. Dee, "Aberystwyth leaf evaluation dataset: A plant growth visible light image dataset of arabidopsis thaliana," *Zenodo*, p. 168 158, 2016. DOI: 10.5281/zenodo.168158.

[100] A. Bender, B. Whelan, and S. Sukkarieh, "A high-resolution, multimodal data set for agricultural robotics: A ladybird's-eye view of brassica," *Journal of Field Robotics*, vol. 37, no. 1, pp. 73–96, 2020. DOI: 10.1002/rob.21877.

[101] D. T. Demie, T. F. Döring, M. R. Finckh, W. van der Werf, J. Enjalbert, and S. J. Seidel, "Mixture × genotype effects in cereal/legume intercropping," *Frontiers in Plant Science*, vol. 13, 2022, ISSN: 1664-462X. DOI: 10.3389/fpls.2022.846720.

[102] S. Seidel, T. Gaiser, T. Kautz, S. Bauke, W. Amelung, K. Barfus, F. Ewert, and M. Athmann, "Estimation of the impact of precrops and climate variability on soil depth-differentiated spring wheat growth and water, nitrogen and phosphorus uptake," *Soil and Tillage Research*, vol. 195, p. 104 427, 2019. DOI: 10.1016/j.still.2019.104427.

[103] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134. DOI: 10.1109/CVPR.2017.632.

[104] J. Li, J. Jia, and D. Xu, "Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks," in *Proc. of the IEEE 37th Chinese Control Conference (CCC)*, 2018, pp. 9159–9163. DOI: 10.23919/ChiCC.2018.8482813.

[105] P. L. Suárez, A. D. Sappa, B. X. Vintimilla, and R. I. Hammoud, "Image vegetation index through a cycle generative adversarial network," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1014–1021. DOI: 10.1109/CVPRW.2019.00133.

[106] Q. Dai, X. Cheng, Y. Qiao, and Y. Zhang, "Crop leaf disease image super-resolution and identification with dual attention and topology fusion generative adversarial network," *IEEE Access*, vol. 8, pp. 55 724–55 735, 2020. DOI: 10.1109/ACCESS.2020.2982055.

[107] H. Nazki, J. Lee, S. Yoon, and D. S. Park, "Image-to-image translation with gan for synthetic data augmentation in plant disease datasets," *Smart Media Journal*, vol. 8, no. 2, pp. 46–57, 2019. DOI: 10.30693/SMJ.2019.8.2.46.

[108] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

[109] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1335–1344. DOI: 10.1109/CVPR.2018.00145.

[110] J. Lin, Y. Xia, T. Qin, Z. Chen, and T.-Y. Liu, "Conditional image-to-image translation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5524–5532. DOI: 10.1109/CVPR.2018.00579.

[111] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. DOI: 10.48550/arXiv.1411.1784.

[112] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 843–852. DOI: 10.1109/ICCV.2017.97.

[113] M. Valerio Giuffrida, H. Scharr, and S. A. Tsaftaris, "ARIGAN: Synthetic arabidopsis plants using generative adversarial network," in *Proc. of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct. 2017, pp. 2064–2071. DOI: 10.1109/ICCVW.2017.242.

[114] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *Proc. of the ACM SIGGRAPH Conference*, ser. SIGGRAPH '22, Vancouver, BC, Canada: Association for Computing Machinery, 2022, ISBN: 9781450393379. DOI: 10.1145/3528233.3530757.

[115] A. Muhammad, Z. Salman, K. Lee, and D. Han, "Harnessing the power of diffusion models for plant disease image augmentation," *Frontiers in Plant Science*, vol. 14, p. 1 280 496, 2023. DOI: 10.3389/fpls.2023.1280496.

[116] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544. DOI: 10.1109/CVPR.2016.278.

[117] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, Cham, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

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

[119] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. of the European Conference on Computer Vision (ECCV)*, Springer, Cham, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

[120] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6023–6032. DOI: 10.1109/ICCV.2019.00612.

[121] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," *arXiv preprint arXiv:1901.09024*, 2019. DOI: 10.48550/arXiv.1901.09024.

[122] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. DOI: 10.1109/TPAMI.2013.50.

[123] A. Oring, Z. Yakhini, and Y. Hel-Or, "Autoencoder image interpolation by shaping the latent space," *arXiv preprint arXiv:2008.01487*, 2020. DOI: 10.48550/arXiv.2008.01487.

[124] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410. DOI: 10.1109/CVPR.2019.00453.

[125] J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, and R. Roscher, "Behind the leaves - estimation of occluded grapevine berries with conditional generative adversarial networks," *arXiv preprint arXiv:2105.10325*, 2021. DOI: 10.48550/arXiv.2105.10325.

[126] Z. Shi, X. Xu, X. Liu, J. Chen, and M.-H. Yang, "Video frame interpolation transformer," *arXiv preprint arXiv:2111.13817*, 2021. DOI: 10.48550/arXiv.2111.13817.

[127] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021. DOI: 10.48550/arXiv.2104.10157.

[128] V. S. F. Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 325–12 334. DOI: 10.1109/CVPR42600.2020.01234.

[129] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. DOI: 10.48550/arXiv.2010.11929.

[130] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" *arXiv preprint arXiv:2102.05095*, 2021. DOI: 10.48550/arXiv.2102.05095.

[131] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021. DOI: 10.48550/arXiv.2103.15691.

[132] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," *arXiv preprint arXiv:2102.00719*, 2021. DOI: 10.48550/arXiv.2102.00719.

[133] C. Ledig, L. Theis, F. Huszár, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4681–4690. DOI: 10.1109/CVPR.2017.19.

[134] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[135] B. Liu, Y. Zhu, K. Song, and A. Elgammal, "Towards faster and stabilized gan training for high-fidelity few-shot image synthesis," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2020.

[136] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015. DOI: 10.48550/arXiv.1511.05644.

[137] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. DOI: 10.48550/arXiv.1511.06434.

[138] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *arXiv preprint arXiv:1807.07543*, 2018. DOI: 10.48550/arXiv.1807.07543.

[139] M. Y. Michelis and Q. Becker, "On linear interpolation in the latent space of deep generative models," *arXiv preprint arXiv:2105.03663*, 2021. DOI: 10.48550/arXiv.2105.03663.

[140] Y. Pang, J. Lin, T. Qin, and Z. Chen, "Image-to-image translation: Methods and applications," *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2021. DOI: 10.1109/TMM.2021.3109419.

[141] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[142] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 465–476.

[143] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2019. DOI: 10.48550/arXiv.1809.11096.

[144] M. Shahbazi, M. Danelljan, D. P. Paudel, and L. Van Gool, "Collapse by conditioning: Training class-conditional GANs with limited data," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2022. DOI: 10.48550/arXiv.2201.06578.

[145] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. of the International Conference on Machine Learning (ICML)*, PMLR, 2017, pp. 2642–2651.

[146] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018. DOI: 10.1609/aaai.v32i1.11671.

[147] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 12 104–12 114.

[148] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016. DOI: 10.48550/arXiv.1610.07629.

[149] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[150] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. DOI: 10.48550/arXiv.1502.03167.

[151] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105. DOI: 10.1145/3065386.

[152] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017. DOI: 10.48550/arXiv.1708.04552.

[153] M. R. Paul, D. T. Demie, S. J. Seidel, and T. F. Döring, "Evaluation of multiple spring wheat cultivars in diverse intercropping systems," *European Journal of Agronomy*, vol. 152, p. 127 024, 2024. DOI: 10.1016/j.eja.2023.127024.

[154] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021. DOI: 10.48550/arXiv.2105.05233.

[155] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. DOI: 10.48550/arXiv.2112.10752. [Online]. Available: https://github.com/CompVis/stable-diffusion.

[156] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. DOI: 10.48550/arXiv.1711.00937.

[157] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021. DOI: 10.1038/s42254-021-00314-5.

[158] Q. Yuan, H. Shen, T. Li, Z. Li, S. Li, Y. Jiang, H. Xu, W. Tan, Q. Yang, J. Wang, *et al.*, "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sensing of Environment*, vol. 241, p. 111 716, 2020. DOI: 10.1016/j.rse.2020.111716.

[159] R. Roscher, M. Rußwurm, C. Gevaert, M. Kampffmeyer, J. A. d. Santos, M. Vakalopoulou, R. Hänsch, S. Hansen, K. Nogueira, J. Prexl, *et al.*, "Data-centric machine learning for geospatial remote sensing data," *arXiv preprint arXiv:2312.05327*, 2023. DOI: 10.48550/arXiv.2312.05327.

[160] X. Specka, D. Martini, C. Weiland, D. Arend, S. Asseng, F. Boehm, T. Feike, J. Fluck, D. Gackstetter, A. Gonzales-Mellado, *et al.*, "Fairagro: Ein konsortium in der nationalen forschungsdateninfrastruktur (nfdi) für forschungsdaten in der agrosystemforschung: Herausforderungen und lösungsansätze für den aufbau einer fairen forschungsdateninfrastruktur," *Informatik Spektrum*, vol. 46, no. 1, pp. 24–35, 2023. DOI: 10.1007/s00287-022-01520-w.

Acronyms

AAE	Adversarial Autoencoder
AE	Autoencoder
CGAN	Conditional Generative Adversarial Network
CGM	Crop Growth Model
CNN	Convolutional Neural Network
CVAE	Conditional Variational Autoencoder
CWGAN	Conditional Wasserstein Generative Adversarial Network
DAP	Days After Planting
DAS	Days After Sowing
DDM	Data-Driven crop growth Model
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
GEM	Growth Estimation Model
GSD	Ground Sampling Distance
IGM	Image Generation Model
IoU	Intersection over Union
IS	Inception Score
KL	Kullback-Leibler
LPIPS	Learned Perceptual Image Patch Similarity
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ME	Mean Error
ML	Machine Learning
MS-SSIM	Multi-scale Structural Similarity Index Measure
PBM	Process-Based crop growth Model

PCA	Principal Component Analysis
PLA	Projected Leaf Area
PSNR	Peak Signal-to-Noise Ratio
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
VAE	Variational Autoencoder
WAP	Weeks After Planting
WGAN	Wasserstein Generative Adversarial Network

List of Figures

List of Tables