UNIVERSITÄT BONN

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

Interpretable Machine Learning for Image-based Harvest-readiness Prediction of Cauliflower

von

Jana Kierdorf

aus
Leverkusen, Deutschland

Bonn 2025

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

Korreferent:
Prof. Dr. Uwe Rascher, Forschungszentrum Jülich, Germany

Tag der mündlichen Prüfung: 18. Dezember 2024

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

Zusammenfassung

Der Anbau von Blumenkohl unterliegt strengen Qualitätsstandards während des Verkaufs, was die Bedeutung eines präzisen Erntezeitpunkts hervorhebt. Die genaue Bestimmung der Erntebereitschaft ist jedoch herausfordernd, da die Blumenkohlköpfe oft von ihrem Blattwerk verdeckt sind. Dies erfordert eine manuelle Ernte, die den Ernteprozess arbeitsintensiv und subjektiv gestaltet. Um diese Herausforderungen anzugehen, steigt das Interesse an der Entwicklung nicht-invasiver, Sensor-basierter Ansätze. Diese bieten schnelle, flächendeckende, kostengünstige und zuverlässige Lösungen, indem sie objektive und nicht-invasive Daten bereitstellen. Die Integration von Zeitreihendaten zur Pflanzenphänotypisierung kann detaillierte Einblicke in die dynamische Entwicklung von Blumenkohl bieten und ermöglicht präzisere Vorhersagen über den optimalen Erntezeitpunkt im Vergleich zu Einzelbeobachtungen. Jedoch ist die Datenerfassung auf täglicher oder wöchentlicher Basis ressourcenintensiv, was die sorgfältige Auswahl der Erfassungstage besonders wichtig macht.

Das Hauptziel dieser Thesis ist die Bild-basierte Schätzung der Erntereife von Blumenkohl. Die Kombination aus Überwachung von Blumenkohlfeldern mit Hilfe von Drohnen und der Anwendung von Deep Learning Verfahren ermöglicht eine automatisierte Schätzung der Erntereife. Allerdings können aufgrund der Feldvariabilität und begrenzter Trainingsdaten Fehler in den Schätzungen auftreten.

Wir bewerten und vergleichen verschiedene Modelle unter Berücksichtigung unterschiedlicher Vorhersagezeiten und Vorhersageziele. Wir stellen ein Framework vor, welches mit Hilfe von interpretierbarer maschineller Lernverfahren die Zuverlässigkeit eines Erntereife-Klassifikators bestimmt. Durch die Identifizierung von Gruppen von Saliency-Maps leiten wir Zuverlässigkeitswerte für jedes Klassifikationsergebnis unter Verwendung von Kenntnissen über die Domäne und die Bildmerkmale ab. Für nicht gesehene Daten kann die Zuverlässigkeit genutzt werden, um (i) Landwirte über Verbesserungen ihrer Entscheidungsfindung zu informieren und (ii) die Vorhersagegenauigkeit des Modells zu erhöhen.

In einem weiteren Ansatz untersuchen wir die Erntebereitschaft basierend auf Zeitreihendaten und analysieren, welche Erfassungstage und Entwicklungsstadien der Pflanzen die Modellgenauigkeit positiv beeinflussen. Dabei verwenden wir die Interpretationstechnik GroupSHAP, um Einblicke in die vorhersagerelevan-

ten Beobachtungstage zu gewinnen und die zukünftige Datenerfassungsplanung zu unterstützen. Durch die Verwendung von Bild-Zeitreihen anstelle einzelner Zeitpunkte erzielen wir eine signifikante Steigerung der Modellgenauigkeit. Group-SHAP ermöglicht die Identifikation von Zeitpunkten, die die Modellgenauigkeit positiv beeinflussen. Durch die Reduktion der Anzahl der Erfassungstermine und die Fokussierung auf diese positiv beeinflussenden Zeitpunkte, verbessert sich die Genauigkeit weiter. Eine selektive Auswahl der Erfassungstage kann somit zukünftig zu einer effizienteren Datenerfassung führen.

Die in dieser Arbeit beschriebene Forschung leistet mehrere bedeutende Beiträge zur Aufgabe der Erntereifeschätzung von Blumenkohl. Sie integriert interpretierbare maschinelle Lernansätze für neuartige Lösungen, um die Klassifikationsgenauigkeit zu erhöhen und ermöglicht es Einblicke in den Entscheidungsprozess der Klassifikatoren zu gewinnen. In der Praxis können diese Einblicke nicht nur zur Verbesserung der Klassifikationsmodelle genutzt werden, sondern auch Landwirte bei ihren Entscheidungsprozessen bezüglich des Erntezeitpunkts und der Datenerfassung unterstützen. Alle Beiträge wurden anhand unseres veröffentlichten GrowliFlower-Datensatzes validiert, der ebenfalls einen wichtigen Teil dieser Arbeit einnimmt und nach dem Peer-Review-Prozess in Konferenzbeiträgen und Fachzeitschriften veröffentlicht. Die Veröffentlichung des Datensatzes unterstützt die Entwicklung und Evaluierung verschiedener maschineller Lernansätze und soll die zukünftige Forschung erleichtern.

Abstract

Cauliflower cultivation is subject to high-quality control criteria during sales, highlighting the importance of accurate harvest timing. However, accurately determining harvest-readiness is challenging because the cauliflower curd is covered by its canopy. This leads to cauliflower being harvested by hand, making the harvesting process labor-intensive and subjective. To address these challenges, there is growing interest in developing non-invasive, sensor-based approaches. These provide fast, field-comprehensive, cost-effective, and reliable solutions by delivering objective and non-invasive data. The integration of time series data for plant phenotyping can provide detailed insights into the dynamic development of cauliflower, enabling more precise predictions of the optimal harvest time compared to single-point observations. However, data acquisition on a daily or weekly basis is resource-intensive, making the careful selection of acquisition days highly important.

The main goal of this thesis is the image-based prediction of cauliflower harvest-readiness. While the combination of monitoring cauliflower fields using drones and applications of deep learning enables automated harvest-readiness estimation, errors can occur due to field variability and limited training data. We assess and compare different models considering different forecasting times and prediction goals. We analyze the reliability of a harvest-readiness classifier with interpretable machine learning. By identifying groups of saliency maps, we derive reliability scores for each classification result using knowledge about the domain and the image properties. The reliability can be used for unseen data to (i) inform farmers to improve their decision-making and (ii) increase the model prediction accuracy.

Another approach examines harvest-readiness based on time series data, analyzing which acquisition days and developmental stages of the plants positively affect model accuracy. We use the interpretation technique GroupSHAP to gain insights into the acquisition days relevant to predictions and to support future data acquisition planning. By using image time series instead of single time points, we achieve a significant increase in model accuracy. GroupSHAP enables the identification of time points that positively affect model accuracy. By reducing the number of acquisition dates and focusing on positively influencing time points, accuracy improves further. A selective choice of acquisition dates can thus lead to

more efficient data collection in the future.

The work described in this thesis makes several significant contributions to the task of harvest-readiness estimation of cauliflower. It integrates interpretable Machine Learning approaches for novel solutions to enhance classification accuracy and gain insights into the classifiers' decision-making process. In practice, these insights can not only be used to improve classification models but also support farmers in their decision-making processes for harvest timing and data collection. All contributions were validated against our published GrowliFlower dataset, which also represents an important part of this work, and disseminated through conference papers and journal articles following the peer review process. The publication of the dataset supports the development and evaluation of various Machine Learning approaches and is expected to facilitate future research.

Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout my thesis journey. First and foremost, I am sincerely grateful to my first supervisor, Ribana Roscher, for her unwavering guidance through all the ups and downs, insightful feedback, and constant encouragement. It has been especially important to me that we have always maintained a friendly relationship, and I felt comfortable approaching her with any issues. Her understanding attitude, even with my numerous injuries and the time commitment required for Korfball, has greatly eased my work during my Ph.D.

I would also like to extend my heartfelt thanks to Uwe Rascher and Lasse Klingbeil. Throughout my Ph.D. journey, we have spent numerous phenotyping courses together, during which I consistently learned something new, also as a co-supervisor. Your presence has always brightened my Ph.D. life, and I am very grateful to have you both on my committee, helping me conclude my Ph.D. journey successfully.

I am immensely thankful to my research group for their collaborative spirit, stimulating discussions, and shared passion for our field. Their camaraderie has made this journey enjoyable and rewarding, both inside and outside the office. Special thanks go to Lukas Drees, my first ever office buddy, who has accompanied me not only throughout my Ph.D. but also since our Bachelor's studies. We share many memories that I will never forget. I also want to extend my gratitude to Timo, my second office buddy for years and the one and only disco fox dancer, to Johannes, our group grumpy yet always humorous colleague, and to Immanuel, who was always there to help and advise me, even when he was no longer there as a Ph.D. student. Thank you all for always being up for a chat, whether you wanted the distraction or not. I hope that the plants in the office will stay alive after I leave.

Other heartfelt thanks go to Birgit Klein and Thomas Laebe, who were my lifesavers in bureaucracy and technical know-how, especially at the beginning. Without you, I would have been overwhelmed in the first year.

Last but not least, my deepest gratitude goes to my family and friends, who have been an invaluable support during this time. Their encouragement, patience, and understanding have provided me with great comfort, and their ability to distract me when needed has helped me maintain my balance. My greatest thanks go to my mother, who supported me throughout and took me in during my long injury phase. Thanks to her, I was able to recover in peace and still continue my Ph.D. studies.

To all of you and also to those not explicitly mentioned here, thank you for being part of this journey. Your support has made this accomplishment possible and unforgettable.

The work presented in this thesis is partially funded by the European Agriculture Fund for Rural Development with contributions from North-Rhine Westphalia (17-02.12.01 - 10/16 – EP-0004617925-19-001) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 491111487 and (RO 4839/7-1 | STO 1087/2-1). In addition, this work was supported in part by the DFG under Germany's Excellence Strategy – EXC 2070 – 390732324.

Contents

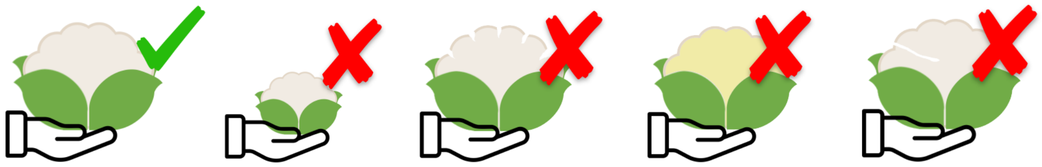
Chapter 1

Introduction

This thesis deals with image-based harvest-readiness prediction of cauliflower using Machine Learning (ML) techniques. We select state-of-the-art ML models to show to what extent image-based predictions are accurate and focus on the use of interpretable ML techniques, which are employed to gain insights into the decision-making process of the learned models. We evaluate trained models using the gained insights by addressing the task of determining the reliability of predictions and the contribution of features to output predictions. We demonstrate that the use of interpretable ML leads to significant improvements in model accuracy for single-input images and time series inputs. Throughout this thesis, we introduce and utilize the dataset that we have collected, processed, and published open source.

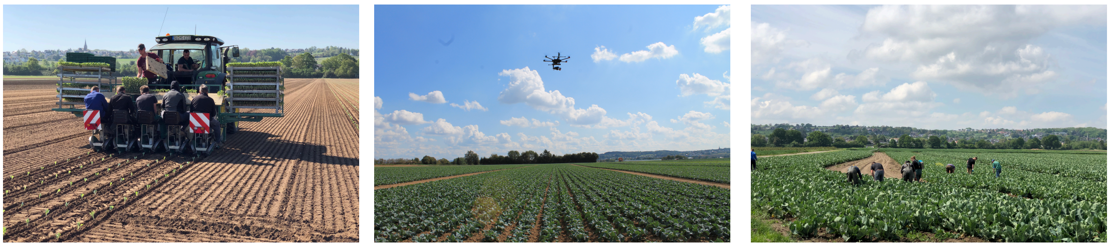
1.1 Motivation

Cauliflower is a suitable target crop plant to develop ML algorithms because its cultivation, morphology, and economic value give rise to many potential applications in the agriculture digitization context. Cauliflower is a high-value crop that must satisfy various quality criteria such as curd size, compactness, color, and overall quality (depicted in Fig. 1.1). Thus, precise timing of plant manage-



(a) Acceptable (b) Size (c) Compactness (d) Color (e) Damage

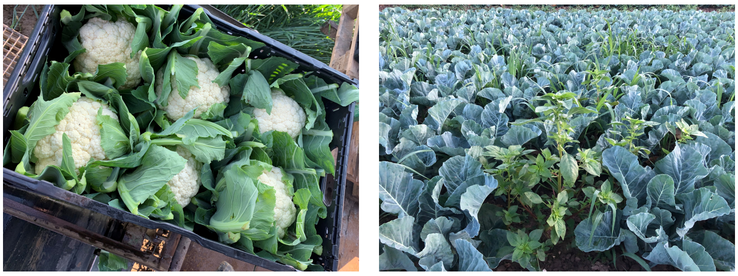
Figure 1.1: Sale requirements for cauliflower. Undersized curds are unsuitable for sale, while overripe cauliflower curds lose their compactness. Poor plant self-coverage results in yellowing curds, further diminishing quality, similarly with damaged plants.



(a) Planting (b) Monitoring (c) Manual harvest

Figure 1.2: Images taken while (a) planting the plants, (b) monitoring the field during growth with a drone, and (c) during a manual harvest run.

ment procedures is required to avoid yield losses due to abiotic or biotic stress and produce marketable cauliflowers. Cauliflower harvesting is labor-intensive because each cauliflower must be harvested within approximately one week, during which the curds are of sufficient size but are not yet overripe. In addition, cauliflower must be harvested by hand due to within-field variability in plant development [1] (Fig. 1.2c). As the curd is covered by leaves as shown in Fig. 1.3b, each individual cauliflower curd must be touched to determine whether it satisfies size criteria subjectively. After cutting and removing the surrounding leaves, product quality is assessed visually to dismiss curds with discolorations, misshapes, or stress symptoms. Note that cauliflower growth is highly dependent on climate, which makes it difficult to predict the most beneficial harvest time. Depending on the prevailing temperature, irradiance, soil water availability, and seed quality and planting depth, plants may develop rather heterogeneously, thus, harvesting of established fields simultaneously can take weeks [2], [3]. Under favorable conditions, plants in sequentially established fields may need to be harvested simultaneously, which requires more workers and lowers the price per cauliflower [4]. Early prediction of harvestable plants and harvest time would facilitate better sales planning and provide significant economic advantages to farmers. Examples of ready and not ready for harvest plants at two acquisition days are visualized in Fig. 1.4. This figure demonstrates that the decision between harvest-ready and not harvest-ready is



(a) Widespread expectation (b) In-field reality

Figure 1.3: The expectation of how a cauliflower plant looks often does not match the reality in the field.

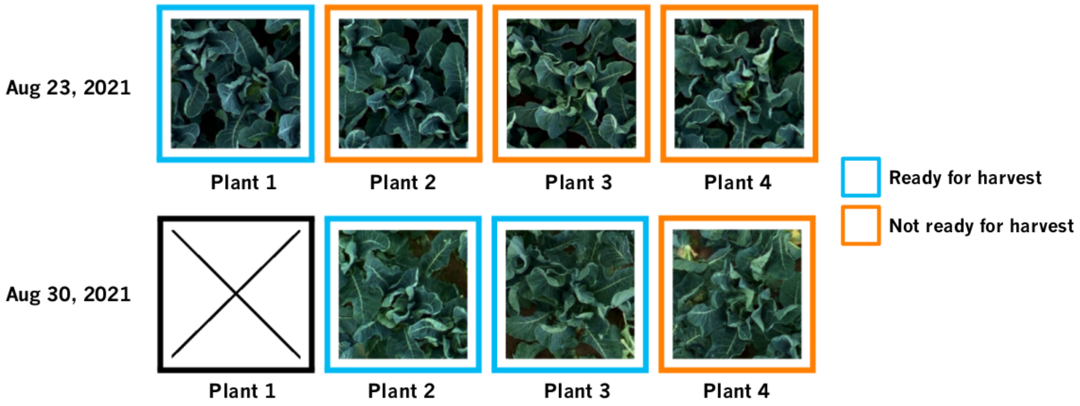


Figure 1.4: Example of `Ready` and `Not-ready` for harvest plants. A column represents the same plant at different times, depicted by the rows. The difficulty of accurately classifying into `Ready` and `Not-ready` for harvest becomes evident in this example due to the similar visual characteristics of the classes simultaneously.

not trivial, highlighting the complexity of accurately determining the appropriate harvest time.

Addressing these challenges requires innovative solutions. Farmers rely on frequent crop monitoring, a time-consuming process requiring expert knowledge, to optimize plant management and support effective decision-making. Typically, farmers and agricultural advisors monitor fields regularly via spot checks of individual plants and extrapolate findings to the entire field. However, this approach is flawed due to individual growth patterns and localized stress occurrences in cauliflower fields. Here, remote sensing and analysis methods can help farmers monitor entire fields more comprehensively [5], [6]. This technology allows for the acquisition of remote sensing data at any scale without damaging or impacting the crops, as exemplified by the use of drones capturing image data, as visualized in Fig. 1.2b. ML methods have become increasingly important in processing and interpreting these large amounts of remote sensing data. ML involves learning a predictive function that relates observations to the desired output, and trained models can be designed flexibly relative to the type of observations. While remote sensing and analysis methods offer a promising way for comprehensive field monitoring, the ability to interpret and understand the outputs of ML models is also crucial. Explainable ML methods provide insights into the underlying decision-making processes of the models, enabling researchers to gain a deeper understanding of crop dynamics. These insights empower researchers to make informed decisions based on model predictions, which can be effectively communicated to farmers to support their decision-making processes.

Furthermore, the development and application of ML methods in agriculture are closely tied to the availability and quality of benchmark datasets with given annotations and in-situ measurements. These datasets are beneficial for advancing

ML methods for plant-specific tasks. However, existing benchmark datasets are often domain-specific and may not adequately address the diverse needs of plant applications. This challenge of generalizing ML models to different plant species but also fields or years showing the same species underlines the need for additional publicly available datasets in plant science. Such datasets are essential for training accurate models across various plant phenotyping tasks, including plant classification, detection, and harvest-readiness estimation. Tracking and analyzing plant development over time is particularly relevant in plant phenotyping, providing valuable insights into plant physiology and growth dynamics.

1.2 Main Contributions

This thesis presents ML-based approaches using interpretation techniques for the classification of harvest-readiness of cauliflower. Explainable ML is employed to assess the reliability of predictions and to determine the contribution of individual acquired data points in the monitored data to model accuracy, thereby enhancing predictive performance. This section summarizes the main contributions of the thesis.

As a first contribution, we introduce GrowliFlower, an agricultural dataset designed for the development of ML approaches. Our dataset focuses on the growth analysis and development of cauliflower plants, facilitating the derivation of phenotypic traits relevant to agricultural applications. The primary objective is to promote advancements in agricultural automation. The open source dataset can be found here: *https://phenoroam.phenorob.de.*

The second contribution of this thesis involves the assessment and comparison of various models for predicting the harvest-readiness of cauliflower. We consider different forecasting times and distinguish between binary classification of harvest-readiness and prediction of harvest days. Through these investigations, we demonstrate how different model approaches vary depending on prediction goals and forecasting time, highlighting which approaches are particularly suitable for accurate predictions of harvest-readiness.

The third contribution deals with the reliability analysis of harvest-readiness estimation. We introduce a framework for deriving a reliability score for classification predictions that operates post-hoc during inference time without human interaction. Thus, the system can be applied to already trained models, requiring no modifications or additional training.

The fourth contribution focuses on the comparison between the analysis of single time points and the integration of time series information showing plant development over time. We demonstrate that models based on image time series data exhibit superior accuracy than those that only consider a single time point as input.

The fifth contribution of the thesis uses the interpretable ML method GroupSHAP to effectively facilitate the selection of time points from time series that contribute highly to the model's prediction and, thus, lead to an improvement of the models. With this information, we selectively determine time points that increase the model's accuracy. We compare the time points with the respective development stages of the plants. From this, we conclude which developmental stages are important to determine harvest-readiness and propose how to reduce data acquisition resources. The findings in the application of cauliflower cultivation can be used to estimate the costs and benefits and determine whether the gain in accuracy justifies acquiring data weeks in advance.

To summarize, the main contributions of this paper are that we

- create an open source dataset, called GrowliFlower, that helps to test and further develop machine learning models;
- compare forecasting times and prediction tasks related to harvest-readiness prediction;
- provide a framework for reliability analysis of cauliflower harvest-readiness classification based on single input images;
- show that the use of time series compared to single time points leads to an improvement in the predictive accuracy of cauliflower harvest-readiness;
- show that using the interpretable machine learning technique GroupSHAP helps to select time points to improve the accuracy further. This information can be connected to growth stages and used to reduce the required resources for data acquisition in future works.

1.3 Organization of the Thesis

This thesis is organized as follows. The work begins with introducing the terminology in Chap. 2 crucial for comprehending the thesis. A detailed description of the related work to this study follows in Chap. 3. The methodology employed in the experiments is described in Chap. 4. The core of the thesis comprises three parts and encompasses the contributions of this work. Part I focuses on the description of the data acquisition and resultant datasets utilized in subsequent parts. In Part II, analyses of harvest-readiness prediction based on single input time points and model reliability can be found, followed by Part III, which delves into investigations of image time series used for harvest-readiness prediction. At last, we conclude our work in Chap. 5 and discuss future work arising from it in Chap. 6.

1.4 Publications

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

- 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`
- J. Kierdorf and R. Roscher, "Reliability scores from saliency map clusters for improved image-based harvest-readiness prediction in cauliflower," *IEEE Geoscience and Remote Sensing Letters*, 2023. DOI: `10.1109/LGRS.2023.3293802`
- J. Kierdorf, 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*, 2024. DOI: `10.3389/frai.2024.1416323`

1.5 Collaborations

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

- A. Emam, M. Farag, J. Kierdorf, L. Klingbeil, U. Rascher, and R. Roscher, "A framework for enhanced decision support in digital agriculture using explainable machine learning," *9th Workshop on Computer Vision in Plant Phenotyping and Agriculture (CVPPA)*, 2024. DOI: `10.13140/RG.2.2.24557.81121`
- N. Penzel, J. Kierdorf, R. Roscher, and J. Denzler, "Analyzing the behavior of cauliflower harvest-readiness models by investigating feature relevances," in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, IEEE, 2023, pp. 572–581. DOI: `10.1109/ICCVW60793.2023.00064`
- J. Kierdorf, J. Garcke, J. Behley, T. Cheeseman, and R. Roscher, "What identifies a whale by its fluke? On the benefit of interpretable machine learning for whale identification," *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 1005–1012, 2020. DOI: `10.5194/isprs-annals-V-2-2020-1005-2020`

- M. Farag, J. Kierdorf, and R. Roscher, “Inductive conformal prediction for harvest-readiness classification of cauliflower plants: A comparative study of uncertainty quantification methods,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 651–659. DOI: `10.1109/ICCVW60793.2023.00072`

- 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, ISSN: 2624-8212. DOI: `10.3389/frai.2022.830026`

There are also publications in which I was involved, which are not part or correlated to the thesis:

- M. Günder, F. R. Ispizua Yamati, J. Kierdorf, R. Roscher, A.-K. Mahlein, and C. Bauckhage, “Agricultural plant cataloging and establishment of a data framework from uav-based crop images by computer vision,” *GigaScience*, vol. 11, 2022. DOI: `10.1093/gigascience/giac054`

- 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,” *Comput. Electron. Agric.*, vol. 190, p. 106 415, 2021, ISSN: 0168-1699. DOI: `https://doi.org/10.1016/j.compag.2021.106415`

- D. Marcos, J. Kierdorf, T. Cheeseman, D. Tuia, and R. Roscher, “A whale’s tail-finding the right whale in an uncertain world,” in *xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, Springer, 2022, pp. 297–313. DOI: `10.1007/978-3-031-04083-2_15`

- Y. L. Chong, P. Nachtweide, J. Bauer, *et al.*, “Iterative AI-assisted annotation for visual pollinator identification and quantification in flower-enriched maize,” *Computers and Electronics in Agriculture*, 2024 (submitted)

- P. Nachtweide, J. Bauer, J. Kierdorf, L. Chong, R. Roscher, R. Bayati, F. Kurth, T. F. Döring, A. Hamm, and S. J. Seidel, “Classical, audio-image- and video-based pollinator identification and quantification to evaluate biodiversity in agroecosystems,” in *64. Tagung der Gesellschaft für Pflanzenbauwissenschaften e.V. Digital tools, big data, modeling and sensing methods for sustainable and climate smart crop and grassland systems*, Göttingen, Germany, Oct. 2023

- P. Nachtweide, J. Bauer, J. Kierdorf, L. Chong, R. Roscher, T. F. Döring, A. Hamm, and S. J. Seidel, “Classical and image-based pollinator identification and quantification in agroecosystems,” in *Fachforum Bienen und Landwirtschaft Strategiekonferenz*, Berlin, Germany, Jan. 2024 (accepted)

Chapter 2

Terminology

In this chapter, we begin by introducing important notations and terminologies that are essential for understanding this thesis.

2.1 Notation

In this section, we summarize the notation used in this thesis. We denote vectors $\boldsymbol{a} = [a_1, \dots, a_I]$ in bold letters and matrices $\mathsf{A} = [a_{ij}] = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_J]$ with capital letters, where i and j iterate over rows and columns, respectively. Scalars are denoted with simple letters, such as a. Sets are denoted in capital calligraphy style $\mathcal{A}$. The elements of a set can be assembled either into vectors or matrices. The distribution of a set is denoted by $P(\cdot)$. For the set $\mathcal{A}$ it is accordingly represented as $P(\mathcal{A})$. Functions are denoted by small calligraphic letters, such as $\mathscr{a}$.

2.1.1 Supervised Task

In the following, we introduce the notations of the data and define a supervised learning task. We have a training set

$$\{\boldsymbol{x}_t, y_t\} \in \mathcal{T}, \quad n = \{1, \dots, T\} \tag{2.1}$$

with T samples of M-dimensional feature vectors $\boldsymbol{x}_t \in \mathbb{R}^M$ and respective class labels $y_t \in [1, ..., c, ..., C]$. The observations are composed in a $(M \times T)$-matrix $\mathsf{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_T]$, while the labels are summarized in $\boldsymbol{y} = [y_1, \dots, y_T]$. Similarly, we have a validation set

$$\{\boldsymbol{x}_v, y_v\} \in \mathcal{V}, \quad v = \{1, \dots, V\} \tag{2.2}$$

and test set

$$\{\boldsymbol{x}_u, y_u\} \in \mathcal{U}, \quad u = \{1, \dots, U\} \tag{2.3}$$

of U and V samples of M-dimensional feature vectors $\boldsymbol{x}_u, \boldsymbol{x}_v \in \mathbb{R}^M$ within the sets, and $y_u, y_v \in [1, ..., c, ..., C]$ respective class labels. For the supervised learning task, we assume that all labels are given a priori.

We define the supervised task with

$$\tilde{\boldsymbol{y}}_{\mathcal{T}} = f(X_{\mathcal{T}}, \Theta) \tag{2.4}$$

where f maps the input observations $X_{\mathcal{T}}$ to the output labels $\boldsymbol{y}_{\mathcal{T}}$. The objective is to learn a function f such that its predicted output labels $\tilde{\boldsymbol{y}}_{\mathcal{T}}$ match the provided reference labels $\boldsymbol{y}_{\mathcal{T}}$. This is done by optimizing Eq. 2.4 with respect to any differential loss function. Furthermore, f should exhibit similar behavior on $X_{\mathcal{V}}$ and $X_{\mathcal{U}}$ as it does on the learned training data. The function f can exhibit various characteristics, which may vary depending on the context. In the study within this thesis, we employ a non-linear function, specifically a neural network architecture, as described in Sec. 4.2.1.1.

2.1.2 Unsupervised Task

In contrast to a supervised task, an unsupervised task is based on the absence of predefined labels. We define the unsupervised task through

$$q = g(X_{\mathcal{T}}), \quad q = \{1, \ldots, Q\}. \tag{2.5}$$

The function g groups the data samples $X_{\mathcal{T}}$ to Q clusters. Various clustering approaches can be employed to realize the function g. We denote the set of clusters as $\mathcal{Q}$. Subsequently, we group the samples $\boldsymbol{x}_t$ according to the clustering applied by g and denote the resulting cluster assignments as set $\{\boldsymbol{x}_t, q\}$.

2.1.3 Interpretation Task

For the interpretation task, Eq. 2.1, Eq. 2.2, and Eq. 2.3 change to

$$\{\boldsymbol{x}_t, y_t, \boldsymbol{x}_t^{\text{tool}}\} \in \mathcal{T}, \quad n = \{1, \ldots, T\}, \tag{2.6}$$

$$\{\boldsymbol{x}_v, y_v, \boldsymbol{x}_v^{\text{tool}}\} \in \mathcal{V}, \quad v = \{1, \ldots, V\}, \tag{2.7}$$

$$\{\boldsymbol{x}_u, y_u, \boldsymbol{x}_u^{\text{tool}}\} \in \mathcal{U}, \quad u = \{1, \ldots, U\} \tag{2.8}$$

where $\boldsymbol{x}_{(.)}^{\text{IT}}$ denotes the corresponding saliency maps for $\boldsymbol{x}_{(.)}$, generated by the interpretation tool IT.

2.2 Feature Attribution

Feature attribution aims to quantify the significance of input features in influencing model predictions [21]. Therefore, feature attribution metrics and methods play a crucial role in understanding the decision-making processes of ML models. Various methods focus on different aspects of understanding model decisions, which are crucial for interpreting and explaining ML models and improving their performance [22]–[24]. Typically, attribution methods are employed post-hoc to the model training, as their main purpose is to explain how the learned models work. Diverse attribution methods have been proposed including saliency methods (see Sec. 4.3.3), attention mechanisms, and rationale models [25]. When dealing with the term feature attribution, we frequently encounter terms like feature importance, relevance, contribution, or selection. The purpose of this section is to categorize these terms to enhance the overall understanding of the work.

Feature importance is a metric that gives a global view of whether a model uses a feature to make its predictions [26], [27]. It refers to quantitative measures indicating the impact of a specific feature on a model's predictions. A higher score for a feature suggests a greater effect on the model used to predict a certain variable, thereby making it more important for accurate predictions. The term feature relevance is often considered as a synonym for feature importance [28].

Feature contribution is a metric that provides information on how much of the score a specific feature adds or subtracts from particular predictions for a specific data point [29]. A reference value is used for this, which is, for example, the average prediction over all samples. It is, therefore, comparable to a metric at the instance level. Feature contribution can be used to determine whether redundant, irrelevant, or noisy features are present in the dataset.

Feature selection is a method that aims to choose features with high importance or contribution to enhance models or allocate resources effectively. Additionally, feature selection serves the purpose of reducing dimensionality, facilitating more efficient computations [28]. While feature selection can initially enhance model performance, there is a threshold beyond which excessive information removal results in a loss of valuable data, ultimately leading to a reduction in model performance. It is worth noting that feature selection occurs before model (re-)training, distinguishing it from feature attribution methods, which involve the prediction of feature importance, relevance, and contribution and typically take place during or after the model training process.

2.3 Components of Model Evaluation: Accuracy, Reliability, Interpretability, Explainability, and Beyond

The integrity of ML models is crucial to ensure accurate predictions [30]. Erroneous or inaccurate predictions can lead to misjudgments, which adversely affect farmers' yields and profitability. With an accurate and reliable model, farmers can make informed decisions that promote the sustainable use of resources and optimize crop yields. In this thesis, we determine the extent to which a model fulfills its designated task, assess the reliability of a model, and check whether the model meets our expectations. Our focus is on model accuracy, reliability, interpretability, and explainability of predictions, all of which are explained in this section. We emphasize the importance of the mentioned components and point out their correlations with related terms such as bias, uncertainty, reproducibility, plausibility, and robustness. The collective integration substantially facilitating the development, evaluation, refinement, and improvement of the model architecture and predictions [31]. This not only improves model accuracy but also aids in the discovery of new knowledge.

2.3.1 Relevant Components for the Thesis

In this section, we delve into the fundamental components of model evaluation crucial for the success of our thesis on image-based harvest-readiness prediction of cauliflower. We define the significance of accuracy, reliability, interpretability, and explainability in the context of developing evaluable models.

Accuracy

Accuracy serves as a quantitative indicator of a model's performance [32]. In the field of classification, it quantifies the proportion of correctly classified examples relative to the total number of instances, emphasizing the model's effectiveness in correctly distinguishing the class labels [33]. Various accuracy metrics exist, as described later on in Sec. 4.2.2. According to Yin et al. [34] the achieved accuracy affects the trust in the model. It depends, e.g., on the composition and diversity of the training data [35]–[37]. Greater variability within the training dataset facilitates improved generalization to unseen test data. It is mandatory for the successful computation of ML models that the training, validation, and test datasets belong to the same domain. When dealing with out-of-distribution data, models typically exhibit lower accuracy, which poses challenges for generalization [38]. In agriculture, this is because plants of distinct varieties may exhibit differing characteristics, thus potentially confounding the model's ability to generalize across varieties. However, it is important to note that accuracy alone does not guarantee the absence of bias and uncertainty or the presence of reliability. Further measures are required to determine the other components [39].

Reliability

We define reliability as the extent to which a result is deemed correct with a high level of certainty, closely aligning with predicted outcomes. In the literature, reliability is often equated with repeatability or reproducibility [40], which is explained in Sec. 2.3.2. However, in this thesis, it carries different meanings.

Interpretability

Interpretability refers to the degree to which a human comprehends the predictions or decisions made by a model [30], [41]. It involves translating complex aspects of model behavior into a format understandable to humans, thereby enhancing transparency in decision-making processes [12]. The comprehensibility of interpretations is subjective, with effective interpretations being understandable to the majority [42]. Interpretations emphasize how the input is mapped to the output predictions [43]. This may involve methods for determining feature importance and contribution [44], or visualizing saliency and attention through heatmaps [45].

Explainability

Explainability refers to the process where interpretable models and their results are informed by human experiences and domain knowledge [12]. This provides explanations about the decisions or predictions of a ML model given by interpretations [23]. This implies the necessity for results to be interpretable before they can be explained [46]. When interpretations align with domain-specific knowledge, they foster greater trust in the explanations provided [33]. A significant application of explainable ML is dealing with the black-box nature of deep learning models. By investigating the model's internal mechanisms, it is possible to fine-tune parameters to enhance performance and analyze the features that drive the model's decisions, thereby ensuring scientifically sound results. This encompasses the pursuit of results that are not only explainable but also reliable and scientifically consistent. Nevertheless, the correctness of the explanation is independent of the correctness of the prediction [42].

2.3.2 Contextualizing within Model Evaluation

With regard to the key components discussed, it is also important to emphasize the broader context in which they operate. We recognize the importance of contextual components such as bias, uncertainty, reproducibility, plausibility, and robustness. It is important to note that there are additional concepts and terms beyond those covered here that also play a role in the development and evaluation of machine learning models for agricultural applications.

Bias

Bias refers to systematic errors or distortions in data [47], models [47]–[49] or human decisions [50], that can lead to incorrect predictions [47]. It significantly influences how a model interprets and processes data. Bias directly impacts the accuracy of predictions and can affect the reliability of a machine learning model by consistently skewing predictions or outputs in a particular direction [47]. Bias may not be apparent when evaluating overall accuracy but becomes evident when analyzing the model's performance across different subclasses [51]. Balancing accuracy and bias mitigation is essential for building reliable models. A biased model may consistently under- or over-predict certain outcomes, leading to unreliable results. The recognition and comprehension of bias can be facilitated through the utilization of interpretation tools [52].

Uncertainty

Uncertainty encompasses various sources inherent in ML frameworks, including data uncertainty, model uncertainty, and the predictive uncertainty of the model [33], [53], [54]. While it is essential to categorize uncertainty into these three sources for analytical purposes, the distinction is often blurred or intertwined, as highlighted by Gruber et al. [55]. Interpretability can aid in understanding the origins and implications of uncertainty. Moreover, reducing uncertainty can significantly improve the accuracy and reliability of ML models.

Reproducibility

According to Rojat et al. [33] reproducibility, indicates that a model produces results with low variance after repeated training on the same dataset. Reproducibility is closely linked to reliability, as a reliable model often delivers reproducible results [56].

Plausibility

Plausibility, indicating that something appears probable or reasonable and is consistent with known facts, domain knowledge, or experiences, is inherently subjective due to variations among individuals [57]–[59]. In the field of ML, it is crucial to differentiate between data, model, and predictive plausibility. Plausibility depends on the underlying factual basis and experiential knowledge. However, while a plausible assertion can bolster trust in model predictions, plausibility is regarded as more subjective compared to accuracy and reliability. Merely appearing plausible does not guarantee reliability.

Robustness

Robustness denotes the ability of a model to yield consistent outputs regardless of perturbations to its inputs [33], [42], such as adding noise or adjusting the blurriness, brightness, and contrast within images [60]–[62]. The robustness of a model relies on the variability present within the training data [35]–[37]. Augmentations applied to the input data can serve to augment this variability. However, such augmentations remain within the distribution of the data samples, underscoring the importance of establishing a diverse foundation of input data samples. When a model is robust, it positively impacts the accuracy and reliability of the model [56]. A robust model architecture enables it to handle various datasets and conditions, resulting in more precise predictions and consistent performance.

Chapter 3

Related Work

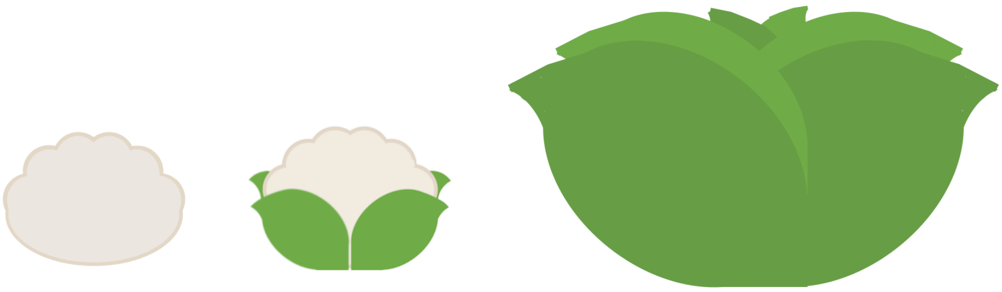
This section covers the existing literature on the research topic. It is intended to provide a comprehensive understanding of the current state of knowledge in this field.

3.1 Cauliflower Harvest Prediction

Predicting plant development and yield is crucial for agricultural planning and management. Recent research, such as by Jin et al. [63], has focused on understanding plant developmental stages. However, unpredictable harvest times and deviations from planned harvest schedules, highlighted in studies like [64]–[66], create challenges for farmers, particularly with cauliflower. Inconsistent harvest timing increases costs and logistical complexities. Lindemann et al. [67] emphasize the financial burden of multiple selective hand harvests, significantly adding to production expenses and requiring extensive planning. Advancements in technology, such as machine learning and remote sensing, now enable researchers to make more informed farm management decisions [68].

For harvest-readiness and yield prediction, a cauliflower plant is usually characterized by three components: *curd*, *head*, and *plant* [69], [70], as shown and explained in Fig. 3.1. Our study adopts these components. Various methods have been explored to predict cauliflower yields and harvest times, focusing on environmental conditions, soil characteristics, agronomic practices, and historical data. Early models use statistical methods, primarily relating cauliflower growth to temperature [67], [70]–[75]. For instance, Wurr et al. [76] examine the relationship between curd diameter and accumulated day degrees, while Olesen and Grevsen [1] model temperature and radiation effects on growth. Later, Rosen et al. [77] develop genome-based models to predict curd induction without relying on temperature data.

Recent approaches include computer vision and remote sensing, particularly



(a) Curd (b) Head (c) Plant

Figure 3.1: Terminology describing components of a cauliflower plant. The term *curd* denotes the predominantly white vegetable structure inside the plant without any surrounding leaves. The term *head* denotes the plant's internal curd, along with a few protective leaves that shield it from external influences. Lastly, when we refer to a cauliflower *plant*, we encompass all organic components visible above the soil level. This encompasses the plant in its natural setting without removing its organic components.

drone-assisted phenotyping, mainly for traditional crops like wheat, barley, and maize [78]. In cauliflower cultivation, initial efforts focused on automating plant localization and harvest-readiness prediction using image segmentation techniques using HSV color space and thresholding [78]–[81]. Grenzdörffer et al. [78] further developed methods to derive geometric properties like crop height and curd diameter. However, these methods require varieties with minimal self-covering growth, a challenge as breeding for stress tolerance increases self-covering, complicating visual derivation of geometric curd properties.

Statistical models primarily predict initial curd induction, while image analysis focuses on crop localization. However, there is a gap in research on continuous harvest-readiness throughout the entire harvest window. In related crops such as broccoli, previous studies have addressed the prediction of the first harvest day [67], [75] and even harvest-readiness [82]. However, fewer challenges exist regarding broccoli curd visibility in these studies compared to cauliflower curd visibility. Our work addresses this gap by performing image-based harvest-readiness prediction for cauliflower varieties with self-covering leaves, where traditional geometric property derivation is infeasible.

3.2 Field Monitoring using Remote Sensing

Field monitoring using remote sensing leverages advanced technologies to enhance the efficiency and accuracy of crop monitoring, offering farmers insights into field conditions without the limitations of traditional methods. Large-scale observations from satellites or aircraft and medium-scale observations from unmanned aerial vehicles (UAV) provide overviews of larger agricultural areas [83]–[85]. Large-area sensor-based crop monitoring detects heterogeneity in the field and supports the farmer's decision-making regarding field management. With such detailed, area-wide information on biotic and abiotic stress, these factors can be counteracted more selectively to support environmentally friendly plant management. Medium-scale and close-range observations acquired from UAVs and ground robots are beneficial for collecting detailed information and are ideal for phenotyping individual plants. For example, Nock et al. [86] use optical remote sensing data to define various traits, e.g., structural and phenotypical characteristics at all levels, from individual plants to large areas. Other applications using remote sensing data include yield estimation [87], yield forecasting [88], and monitoring rapid land surface changes [89].

3.3 Machine Learning in Remote Sensing

Machine Learning methods have become increasingly important [90] in processing and interpreting large amounts of remote sensing data. ML involves learning a predictive function that relates observations to the desired output, and trained models can be designed flexibly relative to the type of observations [91], [92]. For instance, through the application of ML techniques, remote sensing data can facilitate the identification of plant traits [93], [94]. Additionally, ML has been instrumental in large-scale crop type and land cover classification [95]–[97] using time series. A main application area is plant phenotyping, which can be made more objective and automated using advanced ML methods, e.g., deep neural networks. For example, Romera et al. [98], Ren et al. [99], and Scharr et al. [100] trained ML models to infer various phenotypic traits, e.g., the number of leaves per plant. Similar traits can also be derived using a combination of object and leaf keypoint detection, which facilitates observation of plant growth as done by Weyler et al. [101]. Sa et al. [102] employed deep convolutional neural networks to detect single fruits, which served as a precursor for subsequent autonomous harvesting [103]. Drees et al. [16] used time series image data of cauliflower and broccoli to predict field growth using conditional generative adversarial networks [104]. They generated an image of a plant later and employed the Mask R-CNN [105] to calculate the projected leaf area. Another typical agricultural application is field weed control, where weeds, crops, and soil must be distinguished accurately. Using neural networks, promising results have already been achieved, where the task can be approached using classification [106], detection [107], or semantic segmentation [108], [109] techniques.

3.3.1 Interpretability and Explainability

Interpretability and explainability have become a focus in current ML research, with increasing application in plant science. A recent review by Mostafa et al. [110] provides a comprehensive summary of previous research on explainable deep learning in plant phenotyping, offering valuable insights for this thesis. This study highlights how deep learning models can reveal new plant traits, improving the efficiency and accuracy of plant phenotyping solving tasks such as plant species and disease classification [111], [112], counting or segmentation of different plant parts as leaves and yield [113], [114], and generating synthetic plant data [115], [116].

Interpretation ML techniques, like feature attribution methods, are crucial for identifying features that contribute to or are important for yield prediction, phenotypic trait analysis, and plant disease classification [117], [118]. The authors point out that explainable ML clarify the outputs of these models, offering opportunities to enhance them further [119]. In disease diagnosis, these techniques reveal which leaf characteristics indicate disease presence [111], [112], [120], thereby improving detection systems, thereby improving detection systems. For this purpose, straightforward layer visualizations or saliency mapping techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) [121] or Local Interpretable Model-agnostic Explanations [122] are utilized. For yield prediction, they identify key environmental factors and agronomic practices influencing crop yield [123], optimizing agricultural strategies. By analyzing the importance of features like soil fertility, irrigation levels, and weather conditions, researchers can optimize agricultural management strategies to maximize productivity and sustainability [124], [125]. In phenotypic trait analysis, interpretation techniques quantify the contributions of various traits to plant performance, aiding in crop improvement [123], [126]–[128]. By assessing the importance of features like plant height, leaf size, and flowering time, researchers can identify genetic markers associated with desirable traits and inform breeding programs for crop improvement. In collaboration with Penzel et al. [11], we investigate the feature relevance in cauliflower harvest-readiness classification by analyzing the causal relationships of features. Interpretation techniques elucidate which model components identify specific features within input images, essential for accurate classification [129], [130].

To the best of our knowledge and based on the summary provided in this review, it becomes obvious that few works exist in the domain of Deep Learning (DL) and explainable ML that deal with temporal image data. The application of explainable ML to image time series has predominately been performed for satellite data so far due to challenges in time series analysis such as missing time series data, handling equidistant intervals between time points [131], [132], or unequal time series length. Thus, most studies using explainable ML focus on one-dimensional time series data only [33], [133]–[136], such as determining the importance of features in temperature or torque sequences [137]. Leygonie et al. [138] apply an anomaly detection framework on the image-based cauliflower dataset GrowliFlower [7], using ResNet-18 [139] and Grad-CAM to detect anomalies in time series images. In this thesis, we employ feature selection based on maximum feature contribution from single images within a time series of cauliflowers, determined through Group Shapley values [29]. Our investigation aims to identify critical growth stages that significantly influence the determination of harvest-readiness.

3.3.2 Reliability

In addition to model accuracy, the reliability of ML models is a critical concern, yet this topic has received inadequate attention in agriculture. While various approaches, such as regularization and data augmentation, have been proposed to enhance model prediction reliability [140], [141], most training efforts prioritize accuracy and loss minimization without adequately addressing quantitatively determination of reliability or its improvement.

Recent approaches for enhancing and evaluating this reliability use the application of explainable ML [31], [142]–[145]. The capability of explainable ML to ensure transparency in the decision-making processes of ML models enables the validation of their reliability. For example, Kailkhura et al. [146] propose a framework that enhances accuracy, explainability, and reliability ad-hoc, particularly for imbalanced data. Their approach specifically employs an ensemble learning framework, including simple models, and evaluates reliability by quantifying the generalization performance. In contrast, our contribution introduces a framework that derives a reliability score for classification predictions post-hoc during inference, requiring no human interaction. This approach allows for applying pre-trained models without altering their architecture or requiring retraining.

3.4 Plant Science Datasets

Benchmark datasets with annotations and in-situ measurements are beneficial in facilitating the development of ML methods for plant-specific tasks. Various benchmark datasets already exist, however, many of these datasets are domain-specific with highly specific objects, e.g., buildings [147] and animals [148] or other semantics, e.g., land cover [149]. Generally, such datasets are not suitable for plant applications. The link between ML and plant sciences is becoming increasingly important [90], as can be seen from the increasing number of related publications in recent years [109], [150]–[154]. Despite increased demand, to the best of our knowledge, only a few publicly available plant-specific datasets are available for ML purposes.

Among the limited number of publicly available datasets or datasets described in the literature, many were acquired in a greenhouse environment [154]–[157] or are based on synthetically generated data [14], [116], [158], limiting their applicability to real-world scenarios. In particular, the greenhouse-grown plant Arabidopsis thaliana rosettes is frequently used in ML research due to its simple rosette morphology [155]. However, the morphologies of agricultural crop plants are more diverse, and their development is affected by changing environmental conditions and abiotic and biotic stresses. Thus, agricultural datasets that represent real-world field conditions that also cover various challenges, e.g., occlusion, shape variability, pose variability, the colors of plants, and plant parts, are required, such as the datasets [159]–[163]. For cauliflower, a disease dataset was published by Sata et al. [164]. However, the difficulty of generalizing ML models between different plant species remains due to varying plant morphology or the specific objectives addressed within each dataset.

Modeling the temporal development of plant growth and plant traits is an active research area, and this requires datasets that monitor plants over time; however, publicly available time series datasets of plants are rare. One such dataset is the cauliflower (*Brassica oleracea var. botrytis*) and broccoli (*Brassica oleracea var. italica*) dataset from Bender et al. [81]. The data in this dataset were acquired using a camera-equipped robot that captured close-range images at several time points. However, this dataset is limited to only a few plants and lacks semantic information and accurate georeferencing of single plants. This also applies to the Mixed Crop dataset by Drees et al. [16]. The dataset comprises georeferenced time series of RGB images capturing fields of bean-wheat mixtures and reference monocultures, all acquired using drones.

Our contribution is the GrowliFlower dataset, designed for developing ML approaches in agriculture. This dataset facilitates growth analysis, crop development, and the extraction of phenotypic traits for agricultural automation. The dataset includes RGB and multispectral orthophotos of two cauliflower fields throughout the growing period. We provide plant IDs and coordinates for extracting complete and incomplete time series of image patches, along with in-situ reference data like harvest state and plant size. Furthermore, the data set contains image pairs of plants pre- and post-defoliation, enabling analysis of the correlation between external plant appearance and internal curd structure and pixel-accurate labeled data.

Chapter 4

Basic Techniques

In this chapter, we explain methodological backgrounds that are crucial for understanding the thesis and are applied in the research conducted therein.

4.1 Computer Vision Tasks

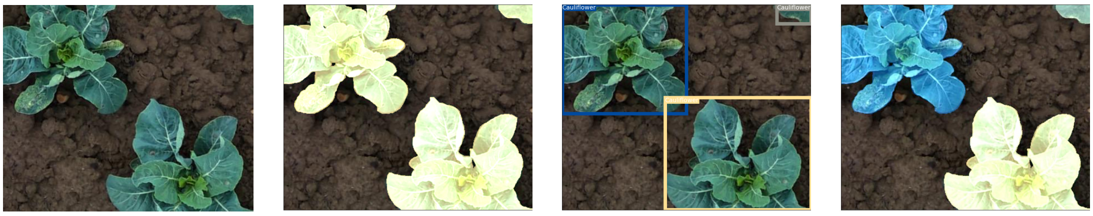
The field of computer vision deals with the automated analysis of large amounts of visual data (e.g., images or videos), employing various techniques for this purpose. Within this thesis, both supervised and unsupervised techniques are applied. The primary distinction between these two approaches lies in the availability of labeled data, which are exclusively available in the supervised task. While the characteristics of the data may differ, our main focus is on image data, placing us within the domain of image processing. Consequently, we offer examples from this specific sub-field in the following.

4.1.1 Supervised Techniques

Supervised techniques enable the exploration of structure and patterns in data using supervision in the form of labels. The disadvantage of supervised techniques is the expertise and time required to label the data. The general notation of the supervised task is specified in Sec. 2.1.1. Supervised techniques can be used for different types of tasks. In the field of image processing, these include classification, semantic segmentation, detection, and instance segmentation.

Image Classification

Image classification aims to categorize an image [165]. The categorization represents either an object, e.g., cauliflower as shown in Fig. 4.1a, or a state, such as determining the harvest-readiness of a cauliflower plant, which can be divided into `Ready` and `Not-ready` for harvest.



(a) Classification (b) Semantic segmentation (c) Detection (d) Instance segmentation

Figure 4.1: Comparison of four supervised learning techniques. While the task of classification (a) categorizes the whole image, semantic segmentation in (b) partitions the image into different semantics, detection in (c) detects and locates objects in images, and instance segmentation in (d) partitions the image into different objects, also from identical semantics.

Semantic Segmentation

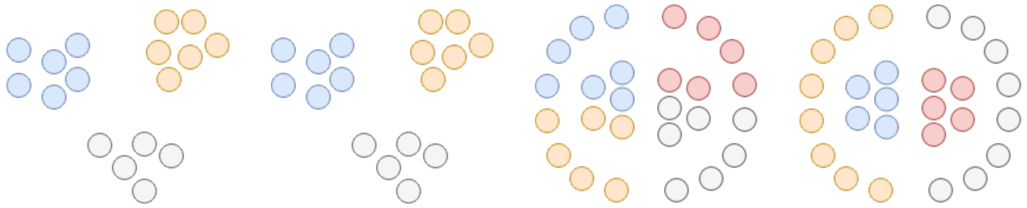
The task of semantic segmentation partitions an image into different semantics [166]. Each pixel is assigned a target class category to generate masks representing various objects or semantics. An application example is the distinction between plants and soil in an image, as illustrated in Fig. 4.1b. All plant pixels are identified and colored uniformly. In semantic segmentation, distinguishing between objects with the same semantics is inherently impossible when they overlap or are adjacent in the image.

Object Detection

Object detection aims to detect and locate objects in images using bounding boxes [167]. The outer frames of the objects are predicted, but not accurate pixel masking is given as shown in Fig. 4.1c.

Instance Segmentation

Instance segmentation combines semantic segmentation and detection in one technique. The aim here is also to partition the image into different instances [105], as shown in Fig. 4.1d. A mask and bounding box are defined for each instance. The difference to semantic segmentation is that different instances of the same semantic are distinguished from each other, even with overlap. In the example of cauliflower plants, this means that different plants are regarded as separate instances, and thus, separate masks are determined. Like detection, instance segmentation enables the localization of objects in images. Furthermore, object properties can be derived using the masks of the individual instances. For instance, in the case of cauliflower, we can determine the diameter of a plant or the projected leaf area.



(a) K-Means (b) Spectral Clustering (c) K-Means (d) Spectral Clustering

Figure 4.2: Overview of clustering techniques.

4.1.2 Unsupervised Techniques

Unsupervised techniques enable the exploration of structure and patterns in data without reference information. No supervision is explicitly required for these techniques, but due to the absence of labels, unsupervised methods are less accurate than supervised methods. The general notation of the unsupervised task is specified in Sec. 2.1.2. In this study, we employ two categories of unsupervised techniques: Clustering [168] and Gaussian Mixture Models [169].

4.1.2.1 Clustering

Clustering techniques group unlabeled data to Q clusters based on similarities, such as the euclidean distance [168]. Consequently, data samples within one cluster are more similar to each other than samples from different clusters. In this chapter, we will focus on two clustering techniques: k-Means and Spectral Clustering, as these will be used throughout the thesis.

K-Means

K-Means is the most basic clustering technique employed with the objective of grouping data $\boldsymbol{x}_t \in \mathcal{T} \cup \mathbb{R}^M$ into a predefined number of Q clusters [170]. The procedure is iterative: initially, Q cluster centers $\mu_q \in \mathbb{R}^M$ with $q = \{1, \ldots, Q\}$ are randomly initialized, and denoted as centroids. Subsequently, the data points $\boldsymbol{x}_t$ are alternately assigned to the nearest centroid based on the smallest distance, followed by an update of centroids according to the new assignments. In this iterative process, the goal is to minimize

$$\sum_{t=0}^{T} \min_{\mu_q \in \mathcal{R}^M} (||\boldsymbol{x}_t - \boldsymbol{\mu}_q||^2). \tag{4.1}$$

K-Means exhibits advantages when applied to uniformly distributed data with comparable cluster sizes, as illustrated in Figure 4.2a. However, challenges arise for clusters that exhibit non-spherical shapes or varying densities [171], as depicted in Figure 4.2c. The outcomes are sensitive to the initialization of the cluster centers, and the presence of outliers can impact the calculated center points, potentially skewing the results toward the outlier. As mentioned earlier, the number of clusters Q must be known initially, although this number can also be determined heuristically.

Spectral Clustering

Spectral Clustering (SC) involves clustering data based on a similarity measure derived from a new representation of the data [172]. The similarities are used as weights in a so-called weight matrix W, from which a graph Laplacian matrix L is obtained. The SC method uses the eigendecomposition of L to determine the clusters using k-Means.

A weight matrix W is constructed from a similarity graph, where we use the Gaussian similarity function based on the Euclidean distance

$$\mathsf{W}_{tm} = \exp\left(\frac{-\|\boldsymbol{z}_t - \boldsymbol{z}_m\|^2}{2\sigma^2}\right) \tag{4.2}$$

for the determination of the similarities. The kernel scale is set to $\sigma = 0.2$ and chosen by evaluating eigenvalues obtained from the weight matrix W. A suitable kernel scale is indicated by significantly different eigenvalues and clear eigengaps. We compute an eigendecomposition of the normalized graph Laplacian [173]

$$\mathsf{L}_{\mathrm{sym}} = \mathsf{I} - \mathsf{D}^{-1/2}\mathsf{W}\mathsf{D}^{-1/2}. \tag{4.3}$$

Here, D is a diagonal matrix, where a diagonal entry is the sum of the weights in the graph for a data sample $\boldsymbol{x}_t$. Note, in SC one uses the eigenvectors $\mathsf{U}_Q := [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_Q]$ for the smallest Q eigenvalues when aiming for Q clusters. In a further step, the rows of U_Q are normalized by

$$B_{tm} = \frac{u_{tm}}{\left(\sum_q u_{tq}^2\right)^{1/2}}. \tag{4.4}$$

to norm 1 and organized in a matrix B. The Q-dimensional vector $\boldsymbol{z}_t$ corresponding to the t-th row of B, for $t = \{1, \ldots, T\}$, gives a new representation for $\boldsymbol{x}_t$ that enhances the cluster-properties in the data [174]. Using the k-Means algorithm, the data are then divided into Q clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_Q\}$ based on the vectors $\boldsymbol{z}_t$.

Notably, SC is not well-suited for scenarios involving numerous clusters. When considering adding a new data point to a cluster, the process necessitates a complete recomputation of the SC analysis or an assignment using the k-Nearest-Neighbors (kNN) method to one of the established clusters. We adopt the kNN approach in our subsequent experiments.

4.1.2.2 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) model complex data distributions using probabilistic approaches. We study the Expectation-Maximization algorithm, a fundamental component of GMMs used for parameter estimation, together with Kernel Density Estimation, which is a non-parametric alternative for density estimation in data analysis [169].

Expectation-Maximization

Expectation-Maximization (EM), developed by Dempster et al. [175], is an algorithm utilized iteratively to estimate the parameters of a probabilistic model that best explains the data. It operates under the assumption that the data originates from a combination of several Gaussian distributions. This algorithm is commonly employed in statistical estimation, especially for tasks such as clustering data or modeling mixed distributions. While theoretically not classified as a clustering algorithm, EM can effectively function as one when the number of Gaussian components is set equal to the number of clusters.

The EM algorithm begins with an initial estimation of the model parameters. It then iterates through alternating ”Expectation” or ”E” step, followed by ”Maximization” or ”M” step. In the E-step, the expected values of the missing or latent variables are computed, including the parameters of the Gaussian distributions. For each data point, the probability is calculated that it originates from each Gaussian distribution of the mixed model. These probabilities are utilized to update the weights of the individual Gaussian distributions for each data point. In the M-step, the model parameters are updated to maximize the likelihood of the data under the model. Typically, the means, covariance matrices, and weights of the Gaussian distributions are updated by adjusting them to represent the weighted data points. The algorithm adjusts the model parameters to best represent the observed data by iteratively performing the E- and M-steps. The E- and M-steps are repeated until the parameters no longer significantly change or a predefined convergence criterion is met.

The EM algorithm is particularly useful in situations where the model is complex. By equating the Gaussian components with the number of clusters, we employ the EM algorithm as a clustering method and analyze it accordingly in the subsequent thesis experiments. Compared to the k-Means algorithm, the EM algorithm identifies and groups clusters with different distributions.

To assign new samples to the distributions, we compute the probability that they originate from each Gaussian distribution within the mixture model. Subsequently, the assignment is determined by the highest probability. This assignment is also termed soft assignment, as a data point can be assigned to one or more

Gaussian distributions with varying probabilities.

Kernel Density Estimation

Kernel Density Estimation (KDE) enables the estimation of the probability density function (PDF) of a random variable based on a sample of data points [176]. For each data point $\boldsymbol{x}$, a kernel function K, commonly referred to as a kernel, is analyzed separately, and subsequently, these individual results are summed to construct the PDF $\widehat{f}$,

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{T} \sum_{i=1}^{T} K_h(\boldsymbol{x} - \boldsymbol{x}_t). \tag{4.5}$$

This approach offers a smoothed representation of the data distribution through the specification of the bandwidth h, serving as a scaling factor. We employ a Gaussian kernel characterized by

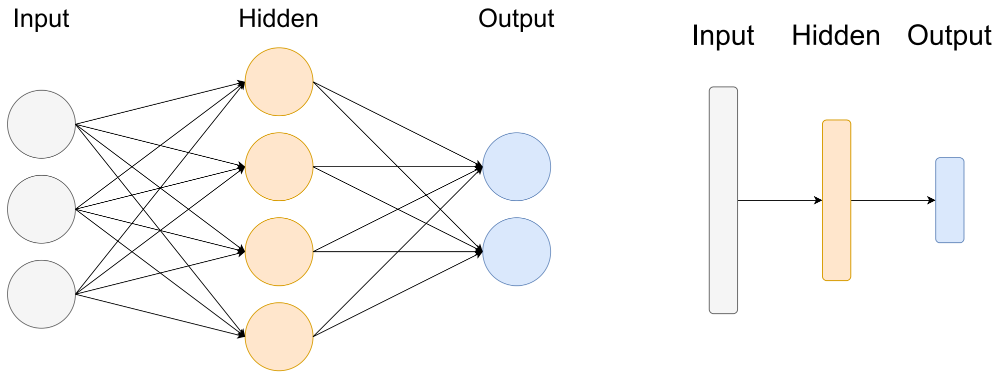
$$K_h(\boldsymbol{d}) = \frac{1}{\sqrt{2\pi h}} \mathrm{e}^{-\frac{\boldsymbol{d}^2}{2h^2}}. \tag{4.6}$$

The selection of the kernel and the bandwidth, which determines the kernels' width, influence the estimation's accuracy. Choosing a too small bandwidth can lead to overfitting, resulting in oversmoothing. Conversely, selecting a too large bandwidth can lead to underfitting, causing undersmoothing, where variations in the distribution are overlooked. KDE is particularly valuable when the underlying distribution of data points is unknown or complex.

To evaluate the goodness of fit of the PDF to the data, we utilize the log-likelihood function as a metric. The log-likelihood function ℓ quantifies the probability of the observed data under the assumption of the PDF.

$$\ell = \sum_{t=1}^{T} \log(\widehat{f}(\boldsymbol{x}_t)) \tag{4.7}$$

A higher log-likelihood suggests that the PDF better fits the observed data, whereas a lower log-likelihood indicates a poorer fit. Consequently, the log-likelihood function is commonly employed to compare various bandwidths or kernels for the PDF and select those that yield the most optimal fits.



(a) Detailed representation showing single neurons.

(b) Generalized representation.

Figure 4.3: Overall arrangement of a neural network (NN). In (a), one circle illustrates one neuron within the network. The arrows indicate weights between neurons of consecutive layers. (b) shows a simplified version of (a) for later visualization purposes.

4.2 Deep Learning

This section describes the networks and evaluation metrics used within this thesis.

4.2.1 Neural Networks

This subsection delves into the fundamental principles of neural networks (NNs). Additionally, we explain two types of NNs, Residual Networks [139] and Vision Transformer [177], in more detail that are employed in this study.

4.2.1.1 Basics

NNs are mathematical models that transform an input into output through a complex and adaptable process defined by an arbitrary differentiable function as described by Eq. 2.4. The overall arrangement of basic NNs is shown in Fig. 4.3. A model consists of a set of neurons organized into multiple layers, where consecutive layers are interconnected by weights W. The output of each neuron, weighted by these connections, serves as the input to the consecutive layer. These weights, along with additional biases $\boldsymbol{b}$, collectively represent the parameters θ of the network

$$\tilde{\boldsymbol{y}} = f(W\boldsymbol{x}) + \boldsymbol{b}. \tag{4.8}$$

Commencing from the input layer, information flows through one or more hidden layers to compute the predicted output scores $\tilde{\boldsymbol{y}}$. This path is called the forward path. The function f is determined by this information flow, and the output $\tilde{\boldsymbol{y}}$ is calculated. The network depicted in Fig. 4.3 can be described mathematically by two functions

$$f(\boldsymbol{x},\theta) = f^{(2)}(W_2 f^{(1)}(W_1\boldsymbol{x})). \tag{4.9}$$

These chain structures are commonly used in NNs. In this case, $f^{(1)}$ represents the function to calculate neurons of the hidden layer, and $f^{(2)}$ is the function to calculate the output. The length of the chain defines the depth of the network.

The iterative optimization of the function f occurs through the minimization of a loss function. Following the forward path, the computed output scores undergo backpropagation along a backward path. This process entails the determination of gradients within the network, enabling adjustments to the weights to minimize the loss function. Cross-entropy (CE) loss is employed for classification tasks in this study.

$$\mathrm{CE} = -\sum_{c}^{C} \boldsymbol{y}_c \log \hat{\boldsymbol{y}}_c, \tag{4.10}$$

with $\hat{\boldsymbol{y}}_c$ as the predicted probability by the model after applying Softmax activation for class c and $\boldsymbol{y}_c$ as the one hot encoded target vector. $\boldsymbol{y}_c$ equals 1 if the sample belongs to class c, 0 otherwise.

NNs are designed to learn the distribution given in the training data. However, they struggle with classifying inputs that fall outside this learned distribution. Therefore, extensive data is essential for training networks capable of generalizing well. To achieve robust generalization, ensuring a diverse range of data inputs is crucial, promoting adaptability across various scenarios.

A variety of neural network architectures exists, among which convolutional architectures hold a dominant position in the field of computer vision, largely due to their exceptional capability in extracting and learning hierarchical features from visual data [105], [139], [165], [178], [179]. Models based on these architectures are referred to as Convolutional Neural Networks (CNNs). The convolutions include inductive biases into CNNs such as translation equivariance and locality, which are used to learn the 2D position in the image and capture spatial neighborhood information. Alternative model architectures like the Vision Transformer incorporate attention mechanisms to capture contextual information within images [177]. Typically, deeper network architectures are better equipped to recognize complex patterns in the input features [139]. However, the increased depth of such architectures gives rise to the vanishing gradients phenomenon [180], where gradients diminish significantly during backpropagation, resulting in slow or stalled learning within deeper layers due to minimal weight updates.

In the following sections, we introduce two model architectures, which are employed in this thesis, Residual Networks in Sec. 4.2.1.2 and Vision Transformer in Sec. 4.2.1.3.

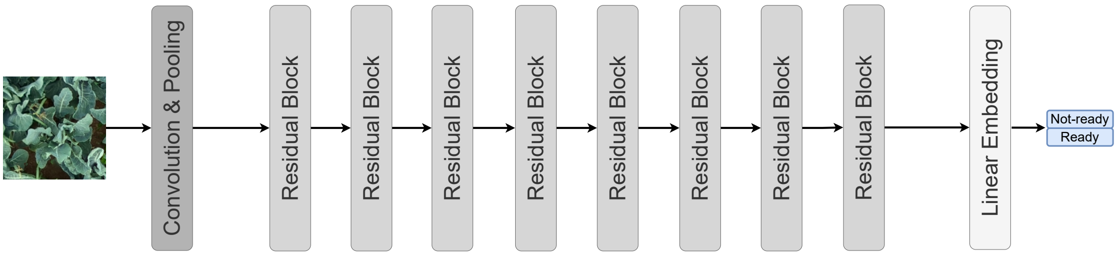


Figure 4.4: Generalized ResNet-18 architecture.

4.2.1.2 Residual Networks

Residual Networks (ResNets) are state-of-the-art ML models for image classification developed by He et al. [139]. These models are designed to circumvent the problem of vanishing gradients using residual connections, even for deep network structures.

Residual Networks for Single Images

A ResNet architecture is characterized by its initial convolutional and pooling layers followed by a series of stacked residual blocks and a linear layer. These residual blocks incorporate convolutional layers and identity shortcuts, also known as skip connections, which enable the network to extract hierarchical features from input images across varying levels of abstraction. The number of residual blocks varies on the specific ResNet architecture employed. In this work, we use a ResNet-18, which consists of two initial layers, 16 convolutional layers organized in residual blocks, and a linear layer for dimension reduction, as visualized in Fig. 4.4.

Residual connections within the blocks facilitate the extraction of features by allowing the network to learn residual functions relative to the input of each block. This mechanism alleviates the issue of vanishing gradients during training, as identified by Glorot et al. [180]. Including residual connections ensures that information from earlier layers is preserved by adding the input of each layer to the output of its corresponding residual block.

After the residual blocks, global average pooling is applied to aggregate spatial information from the final feature map. This pooling operation effectively reduces spatial dimensions while retaining essential features. Finally, a fully connected layer maps the aggregated features to the desired output dimensionality, typically corresponding to the number of classes in classification tasks.

Residual Networks for Time Series

To solve classification tasks based on image time series, we use a siamese network structure inspired by the framework developed by Zapata et al. [181] and combine

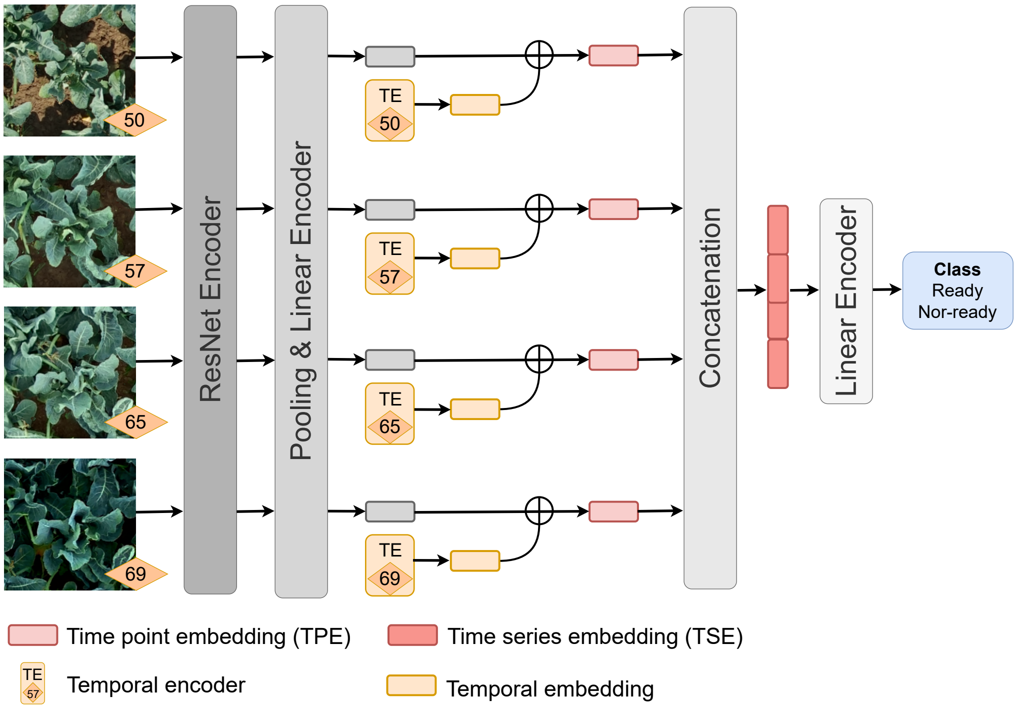


Figure 4.5: ResNet-based architecture for cauliflower image time series classification. Each image of a time series is fed into the network successively. The network weights are updated after the entire time series has passed through the network. Figure source: Kierdorf et al. [9].

the model with a temporal encoding (TE) for contextual relations between time points. The model is visualized in Fig. 4.5.

Given an image time series of length T, each image is sequentially fed into the same ResNet-18 encoder, where the weights are updated only after the entire time series has passed through the network. To obtain a lower-dimensional feature embedding vector that can be used for explainable ML methods (Sec. 4.3.4), we modify the size of the last standard fully connected layer within the encoder to 32. We add a TE of the plant's age to the embedding following the idea of positional encoding in Vision Transformer (ViT). This adjustment facilitates the establishment of contextual relationships among temporal points, thereby enhancing the discrimination between young plants and underdeveloped ones. We refer to the resulting embedding as the time point embedding (TPE). The TPEs are then concatenated to form a time series embedding (TSE), which is fed into a linear encoder consisting of two linear layers to calculate the final scores for each class. The input dimension of the first linear layer in the encoder is equal to the length of the TSE ($T \times 32$). The output dimension is optimized by hyperparameter tuning based on the length T of the time series to retain most of the information. Therefore, the output dimension is defined by dividing the TSE length by a scaling factor λ. We have observed that this additional layer significantly improves the

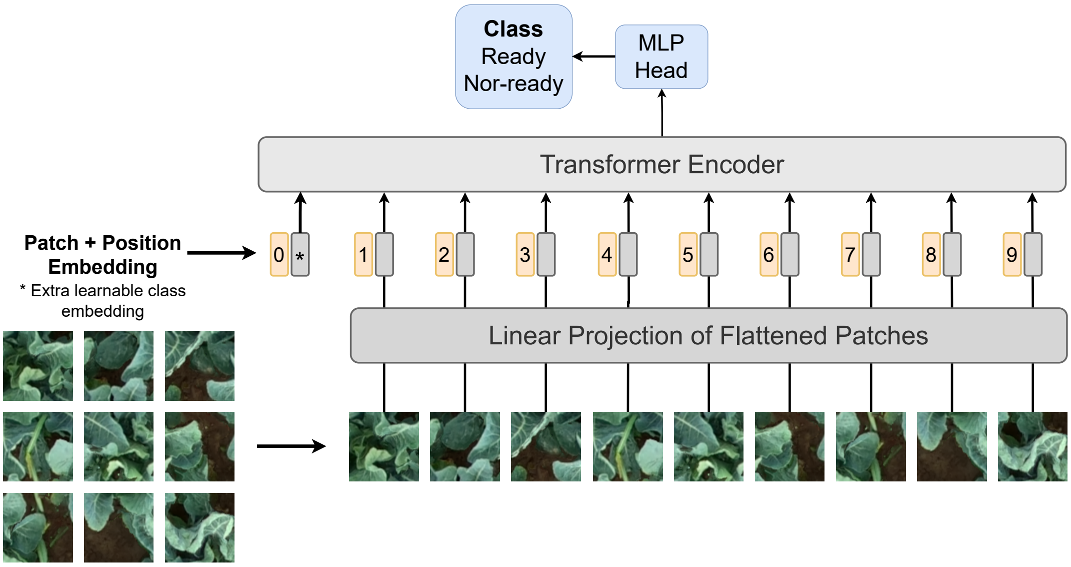


Figure 4.6: Vision transformer architecture. The illustration is based on [177].

classification accuracy for time series.

In our architecture, each time series must have the same length, unlike ViT for time series that can handle varying time series lengths [182]. However, this architecture has the advantage of requiring less data and fewer parameters to train an accurate model.

4.2.1.3 Vision Transformer

The concept of a Transformer originated within the domain from the field of Natural Language Processing (NLP), as introduced by Vaswani et al. [183]. Departing from traditional recurrent and convolutional architectures, Transformers exclusively rely on the attention mechanism. The architecture emphasizes identifying relevant sub-sequences within the input sequence and modeling dependencies among its sequence elements. Consequently, Transformers explicitly encode contextual information, allowing for nuanced understanding and processing of sequential data.

Based on the NLP transformers, Dosovitskiy et al. [177] developed the Vision Transformer (ViT) for analyzing images. We focus exclusively on the application for classification tasks. In the following section, we explain the structure of a ViT for single input images and how we use it for analyzing time series.

Vision Transformer for Single Images

A ViT interprets an image as a sequence of patches, as visualized in Fig. 4.6. To do this, the image is divided into patches of equal size, which are embedded linearly. In addition, a positional encoding (PE) of the patches takes place by calculating a positional embedding for each patch position in the original image, which is then added to the patch embedding. The positional embedding is determined through sinusoidal positional encoding [184]. The encoding is partitioned into both a sine and cosine component, computed according to the following expressions:

$$\begin{aligned} \mathrm{PE}(k, 2i) &= \sin\left(\frac{k}{s^{2i/d}}\right) \\ \mathrm{PE}(k, 2i+1) &= \cos\left(\frac{k}{s^{2i/d}}\right) \end{aligned} \tag{4.11}$$

The position is denoted with k, d is the dimension of the output embedding, s is a user-defined scalar, and i implies the mapping index with $0 < i < d/2$. The positional embeddings do not carry information about the 2D position of patches within the image. The spatial resolution is learned from scratch. That means the model learns to encode the spatial distance within the image through the similarity of the (positional) embeddings.

The embedded patches, alongside an embedded classification token, serve as input to a transformer encoder, comprising multiple stacked multi-head attention layers and multi-layer perceptron (MLP) blocks [183]. The multi-head attention layers play a crucial role in capturing the relation among various patches within the image and computing attention weights among them. This process facilitates the aggregation of contextual information spanning the entire image. In the context of a transformer, the attention distance corresponds to the receptive field size of a CNN. A larger attention distance is similar to a larger receptive field. Unlike CNNs, where convolutions rely on local neighborhood information, self-attention operates globally within the image, enabling information integration across the entire image without any constraints on the distance between pixels.

Following the multi-head attention layer, an MLP is employed to compute a non-linear transformation of the patch representations. Normalization operations are performed between layers, and residual connections are introduced to stabilize the training process of the ViT, similar to the approach utilized in ResNet architectures. The output derived from the transformer encoder constitutes the image representation, which subsequently undergoes processing by a classification head implemented through a MLP to receive the final classification output.

We employ a Vision Transformer with the configuration denoted as ViT-B/16. The nomenclature of the model is determined by its configuration and patchsize. The designation ”B” signifies the adoption of a base model architecture inspired by BERT [185], while the patchsize is specified as 16 px × 16 px.

A ViT often reaches better accuracies for mid-sized datasets than a ResNet. For low-sized datasets, ResNet achieves better accuracies as no inductive biases are given. Transformers need more data to learn, for example, the locality of patch positions.

Table 4.1: Confusion matrix for two classes.

		Prediction	
		`Ready`	`Not-ready`
Reference	`Ready`	TP	FN
	`Not-ready`	FP	TN

4.2.2 Evaluation Metrics

A commonly used method of evaluating the accuracy (Sec. 2.3.1) of neural networks is the confusion matrix. Within the confusion matrix, the reference data is compared to the predictions made by a trained model. An illustration of a confusion matrix for two classes is presented in Tab. 4.1. In this case, we set the two classes to `Ready` and `Not-Ready` for harvest. True positives (TP) indicate the correctly classified samples belonging to class `Ready`, while true negatives (TN) denote the correctly classified samples of class `Not-ready`. False positives (FP) provide information about the number of samples that are incorrectly classified as class `Ready`, while false negatives (FN) indicate the number of samples that are incorrectly classified as class `Not-ready`. As we focus on the harvest-readiness prediction of cauliflower, we focus mainly on this binary decision, but the confusion matrix is also valid for multi-class tasks. Various metrics can be derived from the confusion matrix that are useful for evaluating neural networks. This section provides an overview of these metrics.

Recall

The recall tells us how likely a sample of a given class is classified correctly. It is calculated as follows

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \,. \tag{4.12}$$

The higher the recall, the more individuals are detected.

Precision

The precision indicates the quality of the predictions and is calculated as follows

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \,. \tag{4.13}$$

Greater precision implies that a higher proportion of the identified samples are correctly classified.

F1 Score

The F1 score summarizes and balances precision and recall by calculating a harmonic mean. The F1 score is defined as follows

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \,. \tag{4.14}$$

Overall Accuracy

The overall accuracy (oaAcc) with

$$\mathrm{oaAcc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4.15}$$

gives the proportion of correct classified data samples within a dataset compared to the total amount of data. However, when dealing with imbalanced class data, the oaAcc fails to offer meaningful insights into the performance of individual classes. The method assigns higher importance to classes with larger proportions, masking any potential shortcomings in accurately classifying less represented classes.

Balanced Class Accuracy

Unlike the oaAcc, the balanced class accuracy (bcAcc), also known as the average of recalls, considers class imbalance. It computes the accuracy for each class and then calculates the average across all classes. Eq. 4.16 illustrates the formula for a two-class problem, corresponding to the confusion matrix in Tab. 4.1.

$$\mathrm{bcAcc} = \frac{\frac{\mathrm{TP}}{\mathrm{TP+FN}} + \frac{\mathrm{TN}}{\mathrm{TN+FP}}}{2} \,, \tag{4.16}$$

Intersection over Union

The Intersection over Union (IoU) measures the overlap between the predicted bounding box or segmentation mask and the reference annotation. The IoU is calculated as follows

$$\mathrm{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \,, \tag{4.17}$$

following the evaluation metrics of the COCO dataset [186].

4.3 Interpretability and Explainability in Machine Learning

Explainable ML comprises methods to increase the level of interpretability and explainability to elucidate the decision-making process of ML models. It aids in analyzing complex data by providing visualizations that facilitate the understanding of the data in relation to the model and task at hand [187]. In this section, we delve into the terminology and types of interpretations crucial for categorizing and understanding interpretation techniques. Following this, we describe the specific techniques employed in this study.

4.3.1 Terminology

Before looking at specific interpretation techniques, describing the terminology used for explainable ML approaches is helpful. This aids in determining the type of explanation we require and prefer. Various approaches exist to explain the model's behavior. They are categorized by locality, specificity, and adaptability. Locality encompasses distinctions between local and global approaches, while adaptability distinguishes between post-hoc and ad-hoc approaches. Specificity delineates between model-agnostic and model-specific approaches.

Global approaches have the ability to understand and interpret the whole model behavior across the whole input space [188], [189]. They gain insights into the decision-making process that is applied across all input samples. Global techniques often calculate and provide so-called prototypes, which represent a synthetic sample leading to the maximum score. These prototypes elucidate the features within the input data that contribute most significantly to the model's predictions or decisions.

Local approaches have the ability to understand and interpret one specific model decision [29], [122]. It aims to explain one specific prediction made by the model and allows a human to understand why the model reached a particular outcome.

Ad-hoc or intrinsic approaches are integrated during the training phase [190], [191]. The model is custom-designed to incorporate an interpretable structure analogous to a decision tree, enabling direct interpretability. However, there are still concerns about whether all model features correctly reflect the underlying data or whether some features have not yet been taken into account. By definition, ad-hoc models are inherently model-specific.

Post-hoc approaches are applied after training the model and aim to analyze the learned representations within the model [192], [193], delve into understanding which input activates which neuron in the activation space of the NN, or analyze the reasons for certain classification decisions. Furthermore, post-hoc approaches are adaptable to ad-hoc models after training completion. Sensitivity analyses and feature importance assessments exemplify post-hoc approaches.

Model-agnostic approaches do not rely on specific structures or architectures of the underlying ML model [122], [194]. This has the advantage that no access to the model's architecture is required and forces its application post-hoc. These approaches provide explanations that are applicable across different types of models. They address the effects on the output when the input is perturbed. Model-agnostic approaches are particularly suitable for complex models with numerous parameters, such as neural network, where interpreting all parameters can be challenging. However, they can be applied to any model indiscriminately.

Model-specific approaches rely on specific structures or architectures of the underlying ML model [195], [196]. These approaches give more detailed insights into how the model operates and makes decisions.

In this thesis, we solely utilize local, post-hoc interpretable approaches. This choice is deliberate, as our subsequent experiments involve the investigation of individual plant instance predictions post-training.

4.3.2 Interpretations Types

We distinguish between two different types of interpretations: model-based and decision-based interpretations.

Model-based interpretations focus on understanding internal structures of ML models and how a model processes and transforms the data to output predictions. These interpretations provide insights into the relationship between model components, such as layers, features, and parameters. Model-based interpretations generate prototypes from input data using learned representations that maximize neuron activations and thus maximize the output score for a specific target prediction [197].

Decision-based interpretations indicate why a model makes a specific decision, helping to verify if the model behaves as expected. We distinguish between example-based [198], [199] and attribution-based methods [192], [200], [201]. Example-based interpretations help to highlight specific examples where the model performs well or poorly and how the model behaves for these decisions, while attribution-based interpretations help to investigate the importance or relevance of features driving the model's decision by highlighting the most influencing features.

4.3.3 Saliency Mapping

Saliency mapping refers to the visualization of the attribution of specific features or regions within the input data utilized by a model. It is a straightforward and effective method for visually representing data in an understandable manner [110]. Saliency maps are typically represented in the form of heatmaps, where values within the map are encoded by colors, aiding in identifying patterns, trends, and anomalies compared to numerical matrices. Fig. 4.7 illustrates three exemplary heatmaps (b-d) in conjunction with the input RGB image (a). The appearance of the heatmaps varies, characterized by the color palette chosen, which can be customized according to the task, as well as the underlying methodology employed to determine feature attribution. Further explanations on this topic can be found in Sec. 4.3.4.

In the evaluation of heatmaps, various aspects are considered. Besides the quality of interpretations described by Robnik et al. [42], visual compactness and the extent of highlighted regions are particularly crucial. Strong compactness and a broader range of values within the map contribute to easy interpretability and integration of explainability of the results. Integrating domain knowledge into heatmaps is easier to accomplish compared to numerical data such as probabilities and accuracies. Heatmaps facilitate the determination of object locations and shapes, assessment of data density and compactness, and quantification of the number of elements.

4.3.4 Interpretation Techniques

Machine Learning interpretation techniques aim to explain the model's decision by identifying important regions or specific features within the input. The choice of the visualization tool for interpreting decisions is contingent upon the underlying dataset and the nature of the task at hand. For instance, when the objective is to ascertain the importance of regions or features within an image, attribution-based methods, supplemented by saliency maps, facilitate visualization [192], [202]. In scenarios where the focus is on showing feature contribution to an output prediction of specific known features, saliency maps are less suitable. Still, techniques such as violon plots serve as effective visualization aids [203]. Based on the visualized interpretations, we can predict which regions are important for classifying a specific class or which features contribute most to the output. For cauliflower,

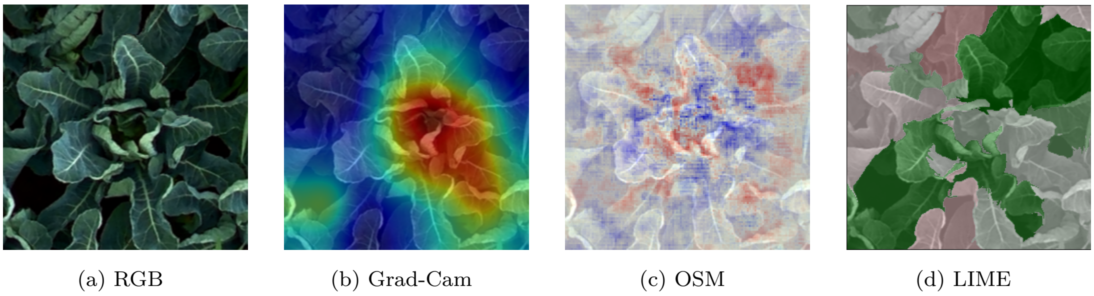


Figure 4.7: Example saliency maps resulting after applying (b) Grad-Cam, (b) Occlusion Saliency Mapping, and (d) LIME on the (a) original RGB image. Image source: Kierdorf et al. [8].

highlighted regions in saliency maps may give an answer about which regions are important for prediction, e.g., `Ready` or `Not-ready` for harvest, or which time points of data acquisition contribute most to high classification accuracy. Combined with domain knowledge, we derive explanations of the interpretation tools' predictions. Various interpretation techniques exist for visualizing the attribution of features to the output. The following techniques are utilized in this study:

Gradient-based techniques leverage the backpropagation algorithm, which calculates gradients of the loss function with respect to model parameters, to understand the contribution of each parameter to the model's output [204], [205].

Perturbation-based techniques modify the input data in various ways and observe the resulting changes in the model's predictions [45], [206].

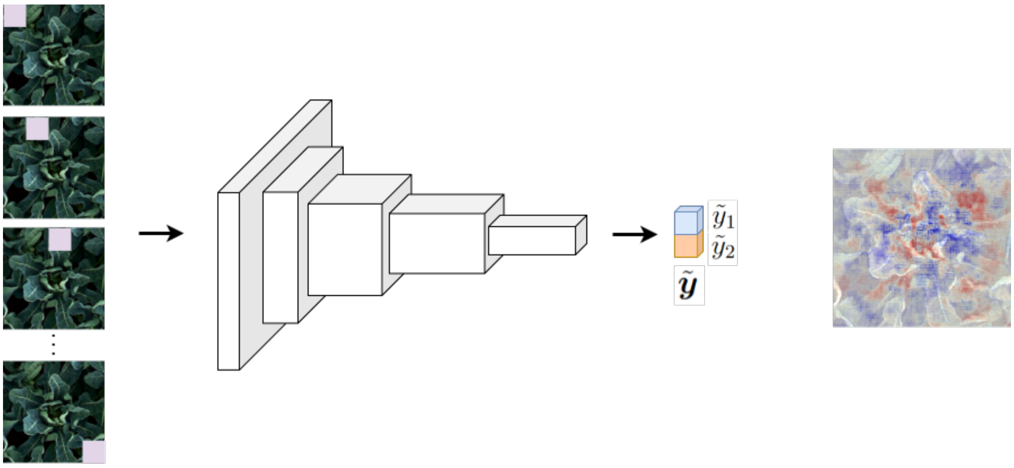
Attention-based techniques compute attention weights to emphasize important input features and to understand feature interactions [189], [207].

Surrogate-model-based techniques involve constructing simpler, interpretable models that approximate the behavior of a complex, black-box model [29], [122]. These surrogate models are typically more transparent and easier to interpret, providing insights into the decision-making process of the original model.

In the following subsections, we delve into four fundamental techniques, namely, Occlusion Sensitivity Mapping, Gradient-weighted Class Activation Mapping, Local Interpretable Model-agnostic Explanations, and Shapley Additive Explanations, providing a detailed explanation as they are employed within this study.

4.3.4.1 Occlusion Sensitivity Mapping

Occlusion Sensitivity Mapping (OSM) is a perturbation-based model-agnostic method developed by Zeiler et al. [45]. This method evaluates the sensitivity of a trained



(a) Overview

(b) Saliency map

Figure 4.8: Visual overview of Occlusion Sensitivity Mapping (OSM) approach shown in (a) and the resulting salience map shown in (b). The purple square overlapping the input images represents the sliding window.

model towards partial occlusions in an image. The use of OSM helps to identify whether the trained model classifies the input based on task-specific features or the surrounding context that is included in the classification. Moreover, it shows which regions contribute positively to the score and which contribute negatively. If the heatmap outcome is a high absolute value at a given pixel position, changing this pixel would significantly affect the classification result. This provides an understanding of the model's learned behavior based on the underlying task.

OSM uses a sliding window approach as visualized in Fig. 4.8 with patchsize p and stride s to permute the input by masking patches and, thus, determine the influence of the occlusion on the predicted model score. The choice of the two parameters influences the result regarding precision and smoothness. In the area occluded by the patch around position o, the classifier's pixel-wise scores for each class are compared to the obtained scores after a part of the image was occluded. The difference $\Delta \boldsymbol{y}_{co}$ is given by

$$\Delta \boldsymbol{y}_{co} = \tilde{\boldsymbol{y}}_c \text{ - } \bar{\boldsymbol{y}}_{co}\,, \tag{4.18}$$

where the original predicted score for each class is denoted by $\tilde{\boldsymbol{y}}_c$, and the predicted score based on occlusion is given by $\bar{\boldsymbol{y}}_{co}$. Performed for the whole image, it results in an occlusion sensitivity heatmap.

Fig. 4.7c shows an example of an OSM heatmap, where blue and red colored areas show the most sensitive pixels towards occlusion. A blue pixel in the map indicates that the score after occlusion is lower than the original score, i.e., this pixel indicates the presence of the examined class. We denote those pixels as positively sensitive. A red pixel indicates that the score after occlusion is higher than the original score, indicating a different class. An occlusion helps predict the correct class. Thus, we denote those pixels as negatively sensitive. The white-colored areas indicate no measured influence on the classification. Note that the smaller s, the finer the map's resolution.

Varying patchsize and stride allows for flexibility in the generation of OSM results. This allows to capture features of different sizes in the image. However, depending on the selected parameters, this increases runtime.

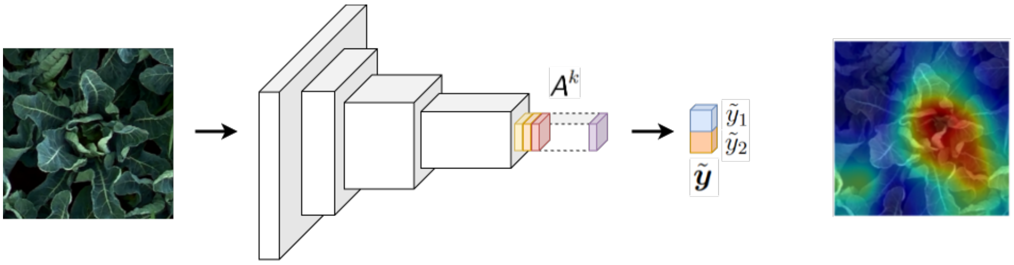
4.3.4.2 Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping (Grad-CAM) is a gradient-based model-specific technique that uses the gradient of the learned network to indicate from which part of an image a given convolutional layer A_k takes information [121]. Eq. 4.19 shows how to compute a Grad-CAM saliency map.

$$\boldsymbol{x}_c^{\text{Grad-CAM}} = \text{ReLU}\underbrace{\left(\sum_k \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \frac{\partial \tilde{\boldsymbol{y}}_c}{\underbrace{\partial \mathsf{A}_{ij}^k}_{\text{gradients via backprop}}} \mathsf{A}^k\right)}_{\text{linear combination}} \tag{4.19}$$

First, the input image is forward propagated through the network to obtain the raw score $\tilde{\boldsymbol{y}}_c$ for the class of interest c. Raw score means the score before the Softmax activation. For this work, the class of interest is the predicted class. All other scores are set to zero. Then, the class-specific gradient $\partial\tilde{\boldsymbol{y}}_c$ of the score for class c is calculated concerning feature maps of a convolutional layer A_{ij}. Exemplary feature maps are visualized in Fig. 4.9 by different colors in layer A_k. Afterward, the gradient of the raw score is backpropagated to the convolutional layer A_k of interest, followed by a global average pooling. Here, Z is the normalization factor that considers the feature map's spatial dimensions with $Z = I \times J$. A weighted combination of activation maps is computed, followed by an activation function like ReLU. An exemplary feature map is shown in Fig. 4.7b.

Gradient-based techniques are particularly effective when discerning objects against smooth backgrounds, such as a bird against the sky, as those techniques primarily focus on detecting edges. Remarkably, these techniques remain effective even in the presence of complicated cauliflower data with numerous edges. Nevertheless, it is important to note that the main objective has shifted from object identification to status classification. While Grad-CAM provides insights into the target class or class of interest, it may not provide information regarding other classes.



(a) Overview

(b) Saliency map

Figure 4.9: Visual overview of Grad-CAM approach shown in (a) and the resulting salience map shown in (b).

4.3.4.3 Local Interpretable Model-agnostic Explanations

Local Interpretable Model-agnostic Explanations (LIME), is a perturbation-based model-agnostic method developed by Ribeiro et al. [122]. LIME perturbs the input and computes the prediction for these perturbed samples with the original model. Perturbation is applied by changing components in images that are meaningful to humans, such as superpixels. After perturbation, a local surrogate model is learned using the perturbed samples. In our work, we use a least squares linear regression model as a local surrogate model. An example saliency map, produced by LIME, is shown in Fig. 4.7d.

While LIME is primarily used for explainability, it can indirectly provide insights into the reliability or fidelity of a model's predictions, as the surrogate model should be a good approximation of local predictions. One disadvantage of LIME is the necessity to pre-select the complexity of the local model. If the underlying model is highly complex or nonlinear, the local approximation may fail to capture the feature details, leading to unstable explanations [208]. Furthermore, in datasets characterized by feature interactions, interpretability is prevented. This limitation is exemplified in scenarios such as distinguishing between a horse and a zebra. Only considering both color stripes together makes sense to define a zebra as a zebra. However, treating them as separate superpixels would worsen their interpretability.

4.3.4.4 Shapley Additive Explanation

Shapley Additive Explanations (SHAP) [29] is a perturbation-based model-agnostic method used to calculate an entity's contribution to a model prediction, where an entity consists of one or more features. The original SHAP approach [29] uses single features, while the GroupSHAP approach [209] considers multiple features within an entity. In our work, we compute GroupSHAP values by defining an entity consisting of a combination of all features within a TPE. Thus, this entity represents the embedding of an input image of a time series. In doing so, we investigate the effect of individual time points on the model's accuracy rather than model features.

In general, an entity with a positive SHAP value contributes positively to a prediction and, thus, increases the model score. In contrast, an entity with a negative value contributes negatively and, thus, reduces the model score. A SHAP value represents the deviation from the mean contribution of an entity to the final prediction. First, all possible entity combinations are formed to determine the SHAP value, where one of these combinations is referred to as a coalition. The entities within a coalition are fixed. Entities not present in a coalition are filled with random examples of the same entity from the training set to maintain a uniform number of entities required for neural networks. Afterward, the SHAP value is determined by computing the mean of differences between all coalitions excluding the entity of interest, compared to the same coalitions, including the entity of interest. We calculate the weighted average over all coalition differences using a similarity measurement of the data samples, e.g., using a kernel function such as Gaussian kernel or binomial coefficients. The resulting value gives the SHAP value for the entity of interest. Coalitions that consist of either only fixed entities or non-fixed entities are given the highest weight, as they are most likely to be used to derive direct entity contributions of the entities of interest. This process is carried out for all entities representing the different TPEs of a time series. The final prediction of a data sample is obtained by adding the SHAP value to the mean prediction of the entire dataset. In general, SHAP values are calculated for each target. One issue to consider when using GroupSHAP is the assumption of entity independence. In real-world scenarios, features are often correlated, leading to misleading interpretations.

Part I

GrowliFlower Dataset

Introduction

Our objective of this thesis is to develop an image-based prediction model for determining the harvest-readiness of cauliflower. However, there is a limited amount of publicly available plant datasets, and none of these datasets capture cauliflower close to its harvest stage, let alone throughout its entire development period. Current research in plant science has not yet advanced to the point where models can be trained to generalize across various plant species. As a result, there is a pressing need for datasets that encompass a diverse range of plant species to support the development of such generalized models.

To close the gap, we present an agricultural dataset and the underlying data acquisition, introduced as GrowliFlower, that is suitable for the development of ML approaches. The proposed dataset is intended to address the growth analysis and development of cauliflower plants and the derivation of phenotypic traits relevant for agricultural applications to promote the development of automation in agriculture. The proposed dataset comprises the following:

- RGB and multispectral orthophotos of two different cauliflower fields were acquired over the entire growing period (from planting to harvest).
- Plant IDs and coordinates, which enables users to extract complete and incomplete time series of image patches showing individual plants accompanied with insitu reference data captured manually on the field.
- The plant IDs and coordinates also allow users to extract image pairs of plants pre- and post-defoliation accompanied with a time series of the respective plant to facilitate analysis of the correlation between the external appearance and internal head of the cauliflower plant..
- The proposed dataset's pixel-accurate labeled data are useful for plant and leaf classification, detection, segmentation, instance segmentation, and other similar computer vision tasks.

We also present two baselines demonstrated application examples of plant and leaf instance segmentation using the proposed dataset in a Mask R-CNN [105] application. The structure, texts and information from this part were mostly taken from our paper written by Kierdorf et al. [7].

Chapter 1

Field Design of the Study Area

In this chapter, we describe the study area. The cauliflower fields used for data acquisition in this study were located on a farm in Western Germany (50°46'6.742" N, 6°58'20.271"O) close to the city of Bornheim, which is 20 km south of Cologne (Fig. 1.1). The mean annual temperature in Bornheim is 14°C, and the mean annual precipitation is 383 mm. This area is dry 142 days a year with an average humidity of 81%. Note that fertile loess soil is available on the farm.

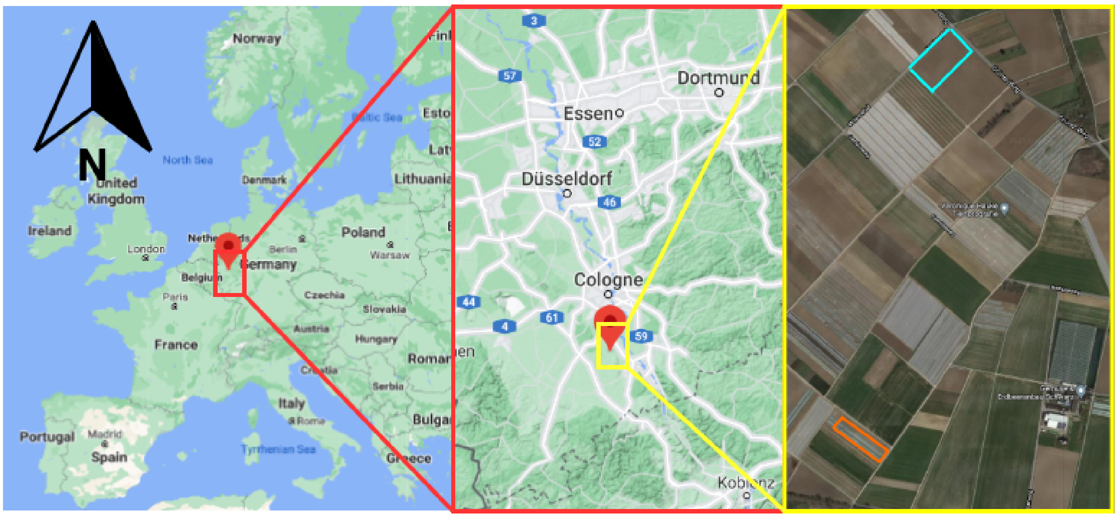


Figure 1.1: Field locations. The fields are located near Cologne, Germany. Blue: field 1 (2020); orange: field 2 (2021). Figure source: Kierdorf et al. [7]. Map source: Google Maps.

We acquired data for three fields, i.e., (1) the field shown in blue in Fig. 1.1 (referred to as field 1 in this thesis) in 2020, and (2) the field shown in orange (referred to as field 2 in this paper) in 2021. Note that the cauliflower plants in fields 1 and 2 were planted in rows in a northwest to southeast orientation. These fields were designed for sprayers with a working width of 18 m. Prior to planting, the fields were plowed to prepare the soil. Tractors with 1.8 m track width were used to plant five rows of nursery-grown young cauliflower plants simultaneously,

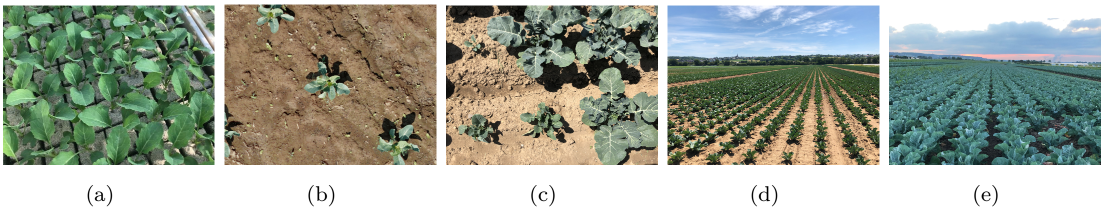

(a) (b) (c) (d) (e)

Figure 1.2: Example field and plant images. Image (a) shows seedling trays prior to planting. Image (b) shows plants two weeks after planting. Images (c–d) were taken four weeks after planting and illustrate how different plants develop over. Image (e) shows plants shortly before head formation. Image source: Kierdorf et al. [7].

with three rows between the tractor tracks. The distance between the rows was 0.6 m, and the distance between the plants in a row was 0.5 m, thereby resulting in a planting density of 33000 plants/hectare. In addition, every 18 m, there was a 2 m wide lane for spraying and irrigation. The fields were subject to conventional farming practices, including hoeing cauliflower plants before canopy closure to reduce weeds and application of pesticides (including herbicides, insecticides, and fungicides). The fields were also irrigated as required using sprinklers. As a result, the abiotic and biotic stresses were rather low in all three fields, and the plants developed rather uniformly.

1.1 Field 1

Field 1 has a width of approximately 100 m and length of 240 m. Thus, total area of field 1 is approximately 2.4 ha. This field was planted with the Korlanu cultivar (Syngenta, Maintal, Germany). Three-quarters of the field were planted using plants from seedling trays (Fig. 1.2, left) on July 28th, 2020 from the southwest direction. The remaining northeastern part of the field was planted on July 29th, 2020. Note that field 1 was generally free of weeds.

1.2 Field 2

Field 2 has a width of approximately 55 m and a length of 210 m. Thus, the area is approximately 1.32 ha. This field was planted with the Guideline cultivar (Syngenta, Maintal, Germany). Here, the plants were transplanted from seedling trays on June 15th, 2021. Note that field 2 contains more weeds than field 1, especially along the southwestern edge of the field due to previous rhubarb cultivation.

Chapter 2

Data Collection

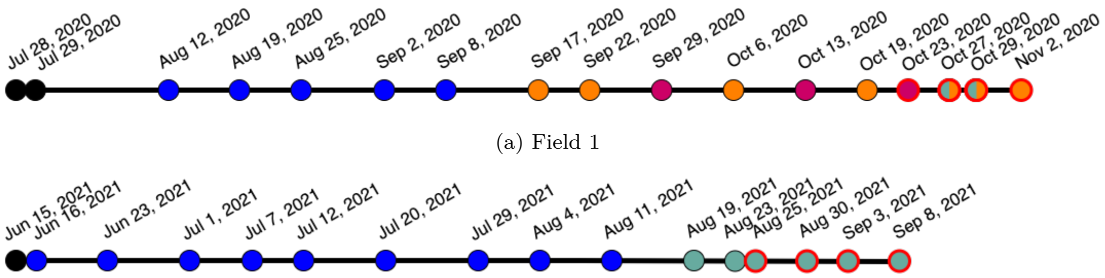
Three types of data were acquired in the data collection process, namely:

1. RGB and multispectral UAV image data with high spatial resolution, which is an indirect measurement of the phenotypic development of the plants.

2. Georeferenced ground control points (GCP) to locate the data in space, spatially arranged according to field size to ensure accurate and robust processing of the orthophotos [210].

3. In-situ measurements of phenotypic traits characterizing the development state and stress factors that serve as reference observations.

The different types of data were collected on the same day to synchronize them. However, to ensure that workers were not visible in the image data, data acquisition processes were not conducted at the same time. Data acquisition was conducted once a week during the entire growth period. During the harvest period, data were collected once between two different harvest days and once after the final harvest. Note that drone flights were only performed on sunny or overcast days to ensure stable illumination for the generation of orthophotos without shading effects due to moving clouds. As a result, the time intervals between successive flights vary. Fig. 2.1 shows the data collection dates for fields 1 and 2. As seen in the top timeline, seven orthophotos are only partly available, which is discussed further in Sec. 3.1. The data collection took a few hours per day, with the in-situ measurements being the most time-intensive. In addition, data collection was adjusted to all field conditions separately, resulting in adaptations to camera settings, number of GCPs, and flight altitude. In the following subsections, we describe the procedure followed for fields 1 and 2.



(a) Field 1

(b) Field 2

Figure 2.1: Timelines of acquired data for (a) field 1 and (b) field 2. The colors represent the data availability for images and in-situ measurements. Figure source: Kierdorf et al. [7].

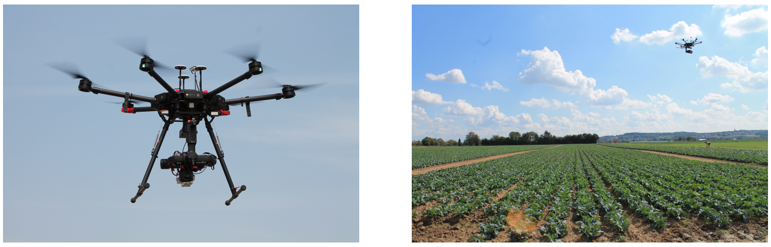


Figure 2.2: DJI Matrice 600 hexacopter for UAV image-based measurements. Image source: Kierdorf et al. [7].

2.1 RGB and Multispectral Imaging

UAV images were captured using a DJI Matrice 600 hexacopter with two mounted cameras (Fig. 2.2). The first camera was a Sony A7 rIII RGB camera with a Zeiss/Batis 2.0 lens (resolution: 47.4 MP). The focal length was 25 mm with a field of view of 71.5°. A shutter speed of *1/1250*th and a *floating* aperture (highest value: 2.0) were selected. The ISO value was set to *automated* for field 1 and changed to 50 for field 2 to align our approach with the image-capture settings recommended by Agisoft. The second camera was a MicaSense RedEdge 3 for multispectral image data. It contains five built-in lenses (resolution: 1.2 MP per band). The wavelengths of the five acquired bands and their respective bandwidth were 475 nm (20 nm), 560 nm (20 nm), 668 nm (10 nm), 717 nm (10 nm), and 840 nm (40 nm). The focal length of the camera is 5.4 mm. For field 1, an altitude of approximately 10 m and an image overlap of 60/80 were used, and for field 2, an altitude of approximately 16 m and an image overlap of 80/80 were used to optimize the data acquisition process and subsequent image data processing. The following factors were considered in terms of the drone flights. For each flight, no irrigation was permitted in or close to the flight area, the drone was flown at temperatures and wind speeds within the device's safe operating range, and the flights only occurred during periods of no rain.

2.1.1 Time Series Flights

On each acquisition date, the drone was flown over a specified area of the field once, which remained the same for the entire growing period. For field 1, this area had a width of 91 m and length of 62 m, resulting in approximately 0.60 ha. For field 2, the area had a width of 30 m and length of 131 m, resulting in approximately 0.39 ha.

The cauliflower plant does not necessarily grow straight, thus, the center of the plant in later growing stages does not match the position of the seedling exactly [78]. Thus, a shift of up to ±10 cm between the center position of the head and the stem position in the early growing stages was observed.

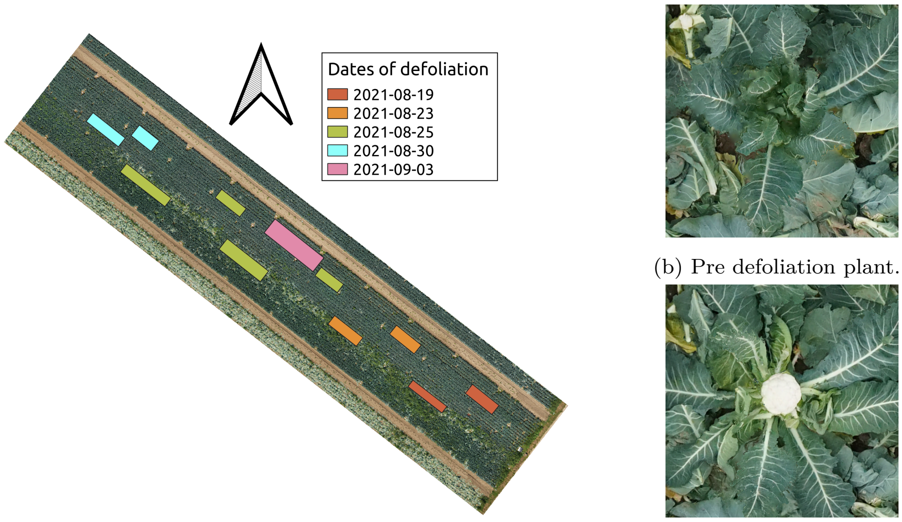
2.1.2 Defoliation Flights

In addition to the time series flights, so-called defoliation flights were conducted. Here, the upper leaf layers covering the cauliflower head were removed manually on individual plants after the time series flight. This step is referred to as defoliation. Note that we ensured that the defoliated leaves did not affect any neighboring plants. The defoliated plants provided information about the development of the head relative to the plant's outer appearance. By performing another UAV flight after defoliation, a dataset of images showing the time series of the plant's outer appearance (Fig. 2.3b) and inner head (Fig. 2.3c) on the day of defoliation was acquired.

For field 1, the defoliation of plants was performed over two days, i.e., October, 27th and 29th, after harvesting occurred. Thus, the defoliated plants represented plants whose head size did not satisfy the harvest criteria, which generally meant that the head was too small. For field 2, starting on August, 19th, when most of the cauliflower heads started developing, between 70 and 200 plants were defoliated weekly. Here, all plants with developed heads were defoliated in rectangular plot regions to minimize the impact of defoliation on the biological growth of neighboring plants. Note that care was taken to not defoliate the reference plants described previously (Sec. 2.3). A distribution of plots for the first five defoliation time points is shown in Fig. 2.3a. For the final flight (after the last harvest), most remaining plants that had not been harvested were defoliated, which resulted in random distribution. Thus, this is not shown in Fig. 2.3a.

2.2 GCP Points

To localize the image data globally in space, the data were georeferenced with the help of circular 12-bit GCPs with a diameter of approximately 20 cm, as shown in Fig. 2.4. Here, the GCPs were fixed in the ground using plastic pegs, and they were



(b) Pre defoliation plant.

(c) Post defoliation plant.

(a) Defoliation plots for field 2.

Figure 2.3: Visual overview of defoliated plant locations for the first five weeks of defoliation in field 2. (b) and (c) show images of a plant pre- and post-defoliation. The locations of randomly distributed defoliated plants from week six are not shown. Figure source: Kierdorf et al. [7].

distributed evenly across the field (refer to Fig. 2 in the appendix) and positioned on tractor tracks or between plants to avoid displacement by external influences, e.g., plowing. In addition, surrounding plants were removed as required to ensure the visibility of GCPs in the image data. We used 21 GCPs in field 1 (35 GCPs/ha) and 44 GCPs in field 2 (113 GCPs/ha) (refer to Fig. 2 in the appendix), with each GCP showing a different pattern. The greater number of GCPs in field 2 was due to the fact that they facilitate subsequent image alignment by ensuring that at least three GCPs were present in each captured image, especially for growth stages with a high degree of plant overlap and dense canopies.

As measuring device for GCP coordinates, a Trimble R4-Model 3 base station with a horizontal standard deviation of ± 5 mm + 0.5 ppm RMS and vertical standard deviation of ± 5 mm + 1 ppm RMS was used for both fields. The measured coordinates were acquired in the WGS84 / UTM 32N coordinate system. To ensure that the markers for the GCPs were not displaced due to external influences, the GCPs were measured at the beginning and end of the data acquisition period to omit displaced GCPs. A third measurement was added for field 2 in the middle of the growing period.

2.3 In-situ Measurements of Plant Developments

In each field, so called reference plots were selected to capture information from reference plants manually. For field 1, four reference plots were assigned (Ap-

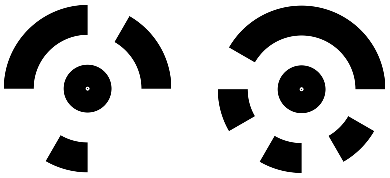
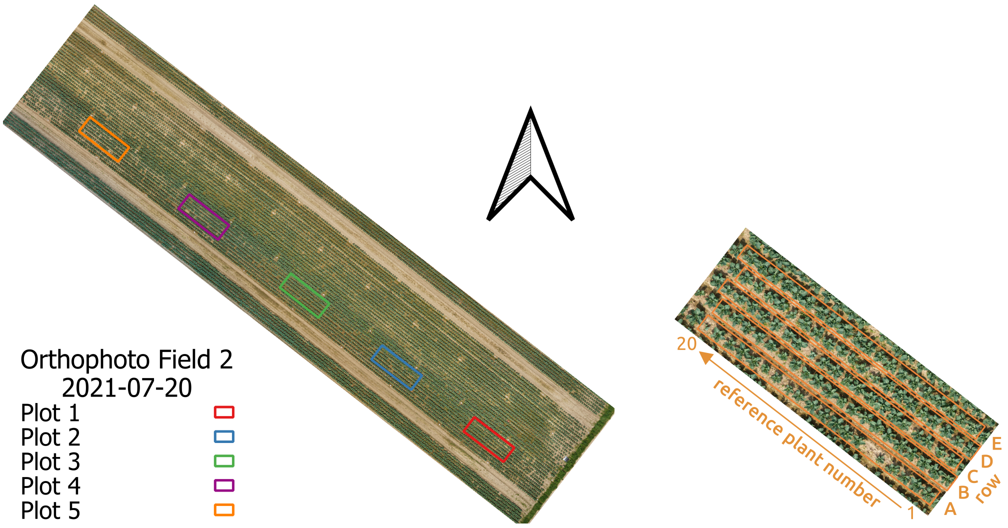


Figure 2.4: Two GCP patterns used for acquisition. Image source: Kierdorf et al. [7].



(a)

(b)

Figure 2.5: Visual overview of (a) reference plots for in-situ measurements in field 2 and (b) the design of reference plot 5 (including reference plants and the ordering of reference plant numbers). The plot design is valid for all reference plots in field 2. Figure source: Kierdorf et al. [7].

pendix Fig. 3a), and each plot comprised three rows with 20 plants each (Appendix Fig. 3b). Thus, each plot contained 60 plants, for a total of 240 plants in all reference plots. The plots were distributed in the northwestern half of the field along the long side. Five reference plots were assigned for field 2 (Fig. 2.5a). Here, each plot comprised five rows of 20 plants (Fig. 2.5b). (100 plants per plot, 500 plants in total). The plots were distributed evenly in the southwestern half of the field along the long side. Thus, the reference data were collected along the entire field. Each reference plant was assigned a specific plant ID identifying the row (field 1: A–C; field 2/3: A–E) and plant number (Field 1: 1–10, 90–99; Field 2: 1–20).

The following measurements were taken for all reference plants in field 1.

1. Phenological development after BBCH according to Feller et al. [211]

2. Height

3. Maximum diameter

4. Other remarks, e.g., stress infestation (listed in the attachment in Tab. 1)

5. Head diameter
6. Harvesting status

Note that the farmer followed a rigorous plant protection schedule, and very few stresses were detected in 2020; thus, information about stresses was not recorded explicitly in 2021. Due to the observed homogeneous development, focus was placed on measurements of BBCH and the height of five representative plants per plot. Here, the head diameter and harvest status were recorded for individual plants.

Available developmental stages of cauliflower are listed in Tab. 2.1. We start with listing stage 12, as the plants were planted in the field out of seedling trays and consist of two or more leaves at the point of planting. The developmental code is made up of the macro stage (first number) and the micro stage (second number). Important stages for cauliflower are macro stage 1 'Leaf development (main shoot)' and macro stage 4 'Development of vegetative plant parts (harvested material)'. We set the mean curd size per harvest day (HD) concerning the day after planting (DAP), illustrated in Tab. 2.2. The colors represent the different developmental stages listed in Tab. 2.1. We see that certain stages of development spread over several flight dates. On average, the harvest-ready plants on different HDs develop at different speeds. Particularly shortly before harvest, major variations between the HDs can be seen. Although the development is spread out, there is a certain correlation between development and acquisition day.

Table 2.1: BBCH developmental stages on the field for cauliflower according to Feller et al. [211]. The code represents the developmental stage and is made up of the macro stage (first number) and the micro stage (second number). The expected curd diameter for cauliflower is about 15 cm. The colors are used to set the code in relation to the acquired data used later on in Tab. 2.2.

Code	Explanation
12	2. leaf unfolds
13	3. leaf unfolds
1x	Stages consecutive to...
19	9 or more leaves unfold
2x	Not available for cauliflower
3x	Developing the main shoot
40	Start of flowering
41	Start of flowering: Vegetation cone width > 1 cm
43	30% of the expected curd diameter is reached
45	50% of the expected curd diameter is reached
47	70% of the expected curd diameter is reached
48	80% of the expected curd diameter is reached
49	Species/variety-typical size and shape achieved; curd still firmly closed

Table 2.2: Overview over the mean curd size per harvest day (HD) per day after planting (DAP). The colors represent the different developmental stages shown in Table 2.1. Additionally, we give the information about days before the first harvest (DBH).

		Mean curd size [cm]			
DAP	DBH	HD$_1$	HD$_2$	HD$_3$	HD$_4$
44	27	0.9	0.7	0.4	0.1
50	21	0.9	0.8	0.5	0.2
57	14	2.1	1.9	1.6	1.2
65	6	7.7	6.1	4.6	3.3
69	2	10.9	8.8	6.7	5.6
71	-	-	12.0	9.4	7.3
76	-	-	-	13.5	10.0
80	-	-	-	-	12.2

Chapter 3

Datasets

The core component of the dataset (Fig. 3.1) comprises both RGB and multispectral orthophotos derived from the captured UAV images. In the orthophotos, single plants are identifiable by their corresponding coordinates and plant IDs. The dataset contains four subsets intended for different ML tasks. The instance segmentation GrowliFlowerL subset contains patches extracted and processed from the RGB orthophotos, and the remaining three subsets contain time series data of individual plants. The GrowliFlowerT subset comprises randomly selected time series data representing a wide variety of cauliflower development. In addition to the time series data, the GrowliFlowerD subset also contains image pairs of plants before and after defoliation. The GrowliFlowerR subset contains the in-situ measurements and the time series data. For each field, a text file containing the measured GCP coordinates at the beginning and the end of field monitoring is provided. For field 2, the GCP coordinates measured during the growing period are also given.

3.1 Orthophotos (GrowliFlowerO and GrowliFlowerM)

The acquired RGB and multispectral UAV images were aligned to orthophotos using the Agisoft Metashape Professional software to obtain a large-scale overview of the monitored fields. Here, the orthophotos were georeferenced according to the measured GCP coordinates. In addition, the individual orthophotos were exported in the WGS84/UTM 32 coordinate system.

The ground resolution for the RGB orthophotos of field 1 is $1.65\,\mathrm{mm\,px^{-1}}$ for the pixel width and height with a minimum and maximum file size of 1.64 GB and 6.7 GB, respectively. The ground resolution for field 2 is $3.10\,\mathrm{mm\,px^{-1}}$ for the pixel width and height with a minimum and maximum file size of 1.3 GB and 5.0 GB, respectively. Twelve orthophotos are available for field 1, where five are entirely

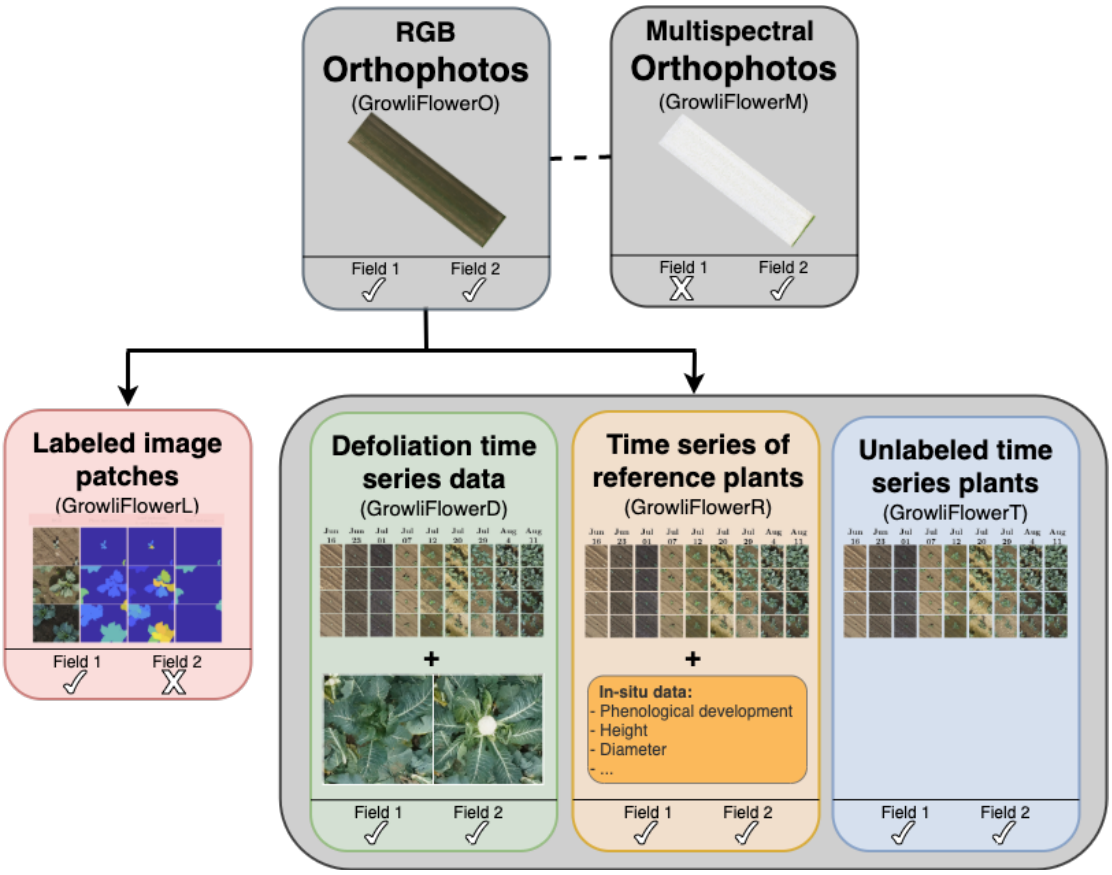


Figure 3.1: Overview of data in proposed GrowliFlower dataset. Figure source: Kierdorf et al. [7].

processed, and seven contain data gaps for small areas where the quality of the UAV acquired images was insufficient. For field 2, 15 orthophotos are available, as shown in Fig. 2.1b. This set of orthophotos is provided in the GrowliFlowerO subset of the proposed dataset. In addition, the dataset contains multispectral orthophotos for field 2 with a ground resolution of 2.5 cm px^{-1} width and length, denoted as the GrowliFlowerM subset.

3.2 RGB Image Patches

In this section, we describe the data extracted from the RGB orthophotos. Note that the ground resolution of the resulting image patches is the same as that of the respective orthophotos.

Each of the following datasets (excluding the labeled dataset described in Sec. 3.2.1) contains a text file with global information for each field, containing the image ID, including the plant ID, and corresponding georeferenced UTM coordinates of the plants. Note that the coordinates identify the center of the plants as observed on August, 19th for field 1 and July, 7th for field 2. In addition, information about the planting day and a proposed assignment as a training, validation, or testing subset is provided as a basis to compare ML methods. To minimize spatial correlation between sets, the proposed training, validation, and testing subsets are spatially disjoint. However, certain systematic factors from a biological perspective are not excluded. The use of these sets is expected to promote the development of ML methods with high generalizability. For the reference data discussed in Sec. 3.2.3, the harvesting time is specified, and for the defoliation data discussed in Sec. 3.2.4, the defoliation date of the plants is specified. In addition, text files with local information for each acquisition date are provided, including the image ID to connect the local information with the global information, and the corresponding local pixel coordinate relative to the respective orthophoto for each data acquisition day. Also, note that information about the day after planting (dap) is included.

To use image patches showing single plants, the patches must be extracted from the orthophotos using the plant IDs and coordinates. Here, an image side length and width of at least 490 px for field 1 and at least 256 px for field 2 is recommended to ensure that the entire plant is captured in the image patch regardless of the plant developmental stage.

3.2.1 Labeled Image Patches (GrowliFlowerL)

This subset, called GrowliFlowerL, comprises pixelwise, manually annotated images, thus, it is well-suited for classification, semantic segmentation, detection, instance segmentation, or stem detection tasks. For this subset, the image patches of four acquisition dates for field 1 are extracted using a sliding window approach. The image patches have a size of 368 px × 448 px. Here, the size of the patches differs from that of the proposed sizes because only plants from earlier development stages are included. In addition, in this dataset, the focus is not on individual plants but on the variability between images, thus, the plants are not located in the center of the patch.

For each RGB image patch, four annotated masks are provided. These annotated masks contain segmentations of (1) plant instances, (2) leaf instances, (3) void segmentations, and (4) stem positions.

(1) The plant instance mask segments the image in soil and plant pixels with instance information for the plants.

(2) The leaf instance mask segments the plants into single leaves. Note that plants at image borders for which no stem or only one-quarter of the plant is visible are annotated as void and no leaf annotation is applied.

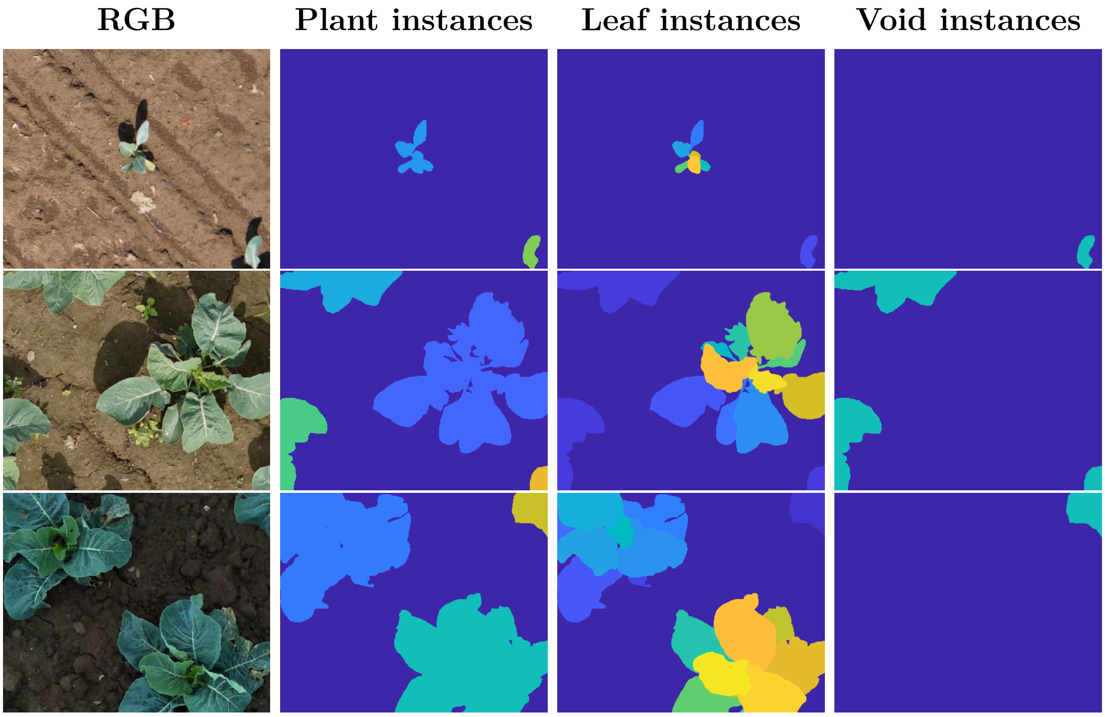


Figure 3.2: Examples of labeled images for different time points. Column 1 shows the RGB base for columns 2–4 which illustrate the corresponding labeled plant instance masks, leaf instance masks, and void segmentation masks. The rows represent different points in time. Dark blue represents the background class, and the other colors represent different (leaf) instances. Figure source: Kierdorf et al. [7].

(3) The void segmentation mask is a binary mask where plants located at image borders where no stem is visible are segmented as void. In addition, plants with only a small amount of visible leaf material in the RGB image are also segmented as void.

(4) The stem annotation mask represents the position of the stems of non-void plants.

Examples of (1) plant instance masks, (2) leaf instance masks, and (3) void segmentation masks are shown in Fig. 3.2. Two things to note are that weed is not labeled as a plant but as a background and that stem positions are only represented by individual pixels, thus, they are difficult to recognize visually. Therefore, masks that include stem information are not shown in these examples. The annotations are provided with a defined name based on the name of the RGB image patch. Here, each patch contains a maximum of four plants, and several patches in the dataset contain no plants (Tab. 3.1). This subset is divided into training, validation, and testing sets, and the complete labeled subset is denoted GrowliFlowerL.

Table 3.1: Overview of distribution of labeled images acquired on different dates.

Definition	All images	Images with plants	Images without plants
		[Train/Val/Test]	*[Train/Val/Test]*
2020/08/12	844	745 *[521/110/112]*	99 *[71/15/15]*
2020/08/19	892	781 *[547/117/117]*	111 *[78/16/17]*
2020/08/25	383	367 *[257/55/55]*	16 *[12/2/2]*
2020/09/08	79	79 *[56/11/12]*	0 *[0/0/0]*

3.2.2 Time Series for Plant Data (GrowliFlowerT)

For each field, the plant coordinates are provided to allow users to extract time series plant images. This data is denoted GrowliFlowerT. The time series data of field 1 comprise the early plant developmental stages and the harvest dates but lack dates when the canopy around the cauliflower head was closed. The time series data of field 2 comprise all growth stages.

For field 1, the coordinates for approximately one-third of the plants are determined (3804 plants in total). The distribution of the location of the extracted data is visualized in Fig. 4a in the Appendix. The selected plants are distributed along the southeastern edge of the field due to the availability of data for most time points and the ability to determine the harvest window of individual plants. The subset is divided into training, validation, and testing sets, as shown in Fig. 4a in the Appendix. In addition, cauliflower planted on July, 28th or July, 29th are included in all three sets to ensure that the variability within the sets is guaranteed. Note that the orthophotos do not overlap entirely; thus, image data are not available for all plants at all times, which results in temporally incomplete time series data. For field 2, 8736 plant coordinates were extracted and distributed evenly over the field. The subset is divided into training, validation, and testing sets, as shown in Fig. 4b in the Appendix. Here, all plant coordinates are provided as georeferenced UTM coordinates.

To use individual plant images, the user must crop the patches around the local plant coordinates determined in the subset. In addition to all global plant coordinates, this subset contains the local coordinates of the patches for each acquisition date, which at a size of 490 px × 490 px for field 1 and 256 px × 256 px for field 2 lie completely within the orthophoto and are not showing spatial data gaps, as patches shown in Fig. 3.5b. Five examples of the time series data are shown in Fig. 3.3 for field 1, and four examples are shown in Fig. 3.4 for field 2. Due to the spatial data gaps, the number of coordinates per date for field 1 varies, which leads to temporal gaps in the time series data. The largest set of time series that includes equal time steps consists of 3611 time series based on eight time points, including the five time points up to day after planting 42

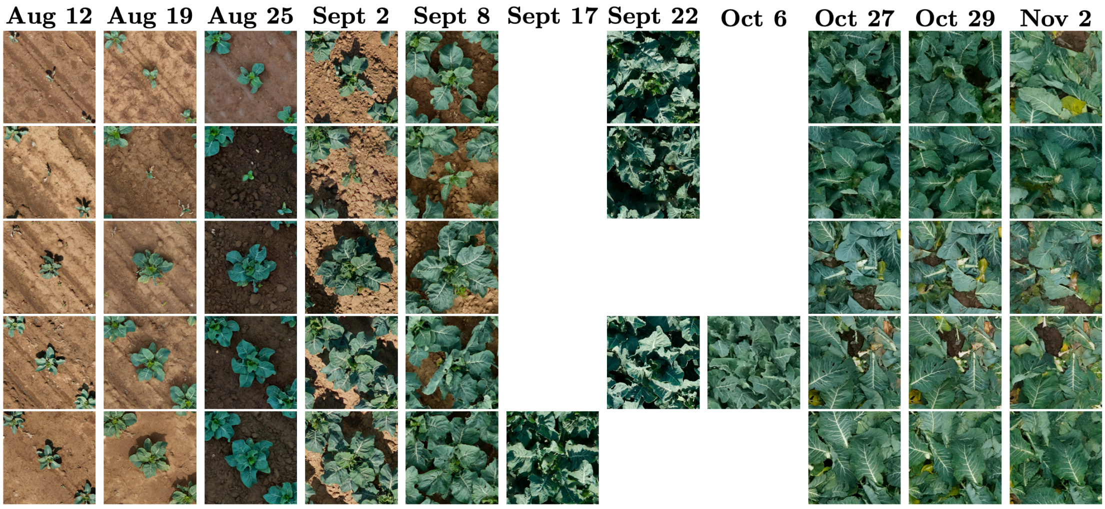


Figure 3.3: Time-series illustration of five different plants in GrowliFlowerT subset in field 1. All rows represent time series of plants containing temporal data gaps due to the poor image quality of the corresponding UAV images (indicated by omitted images). The columns represent the recording days and show the five representative plants captured at the same time on that day. Figure source: Kierdorf et al. [7].

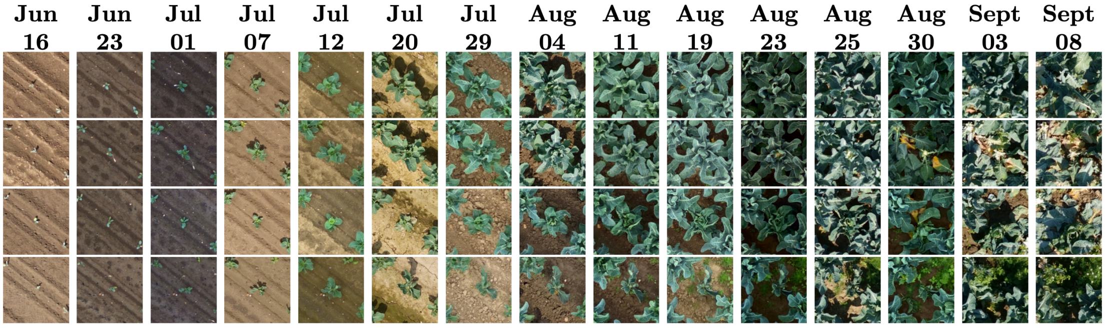
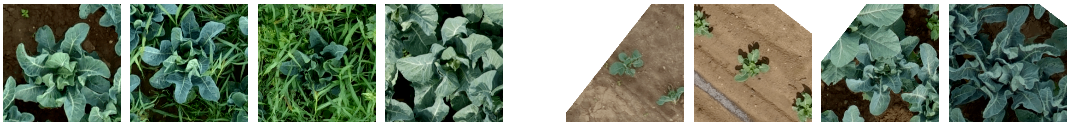


Figure 3.4: Four plant time series from field 2. A row represents a time series. The columns represent the acquisition dates. Figure source: Kierdorf et al. [7].

(Sept, 8$^{\text{th}}$), and all three time points from day after planting 91 (Oct, 27$^{\text{th}}$). In addition to the file that contains all UTM coordinates, a text file containing the UTM coordinates for this set is also provided; thus, the user can extract the time series data for the selected plant IDs. After removing the patches with spatial data gaps, we retained 8348 complete time series images for field 2. Due to the heterogeneous weed occurrence in field 2, the patches contain different amounts of weed, as shown in Fig. 3.5a. Due to the given UTM coordinates, it is possible to extract the complete time series set of local coordinates for both fields if required.



(a) Different amounts of weed occurrence on acquisition date August, 11th.

(b) Data gap occurrence.

Figure 3.5: Data gaps and different amounts of weed occurrences in image data during different stages of growth. Figure source: Kierdorf et al. [7].

Table 3.2: Number of reference plant image patches per acquisition date for field 1 (2020).

Date	Aug 12	Aug 19	Aug 25	Sept 2	Sept 8	Sept 17	Sept 22	Oct 06	Oct 19
# Images	239	239	239	239	239	239	–	–	193
Date			Oct 27 Post			Oct 29 Post			Nov 2
# Images			119			119			12

3.2.3 Time Series for Reference Plant Data (GrowliFlowerR)

For each field, the subset includes the plant IDs and coordinates, which allows the user to extract an image time series set of monitored reference plants that appear similar to those described in Sec. 3.2.2. The time series data for field 1 comprise the early plant developmental stages and the harvest dates, but lack dates, when the canopy was closed. The time series data for field 2 comprise all growth stages (see Fig. 3.4). Tab. 3.2 shows the distribution of available plant IDs and the number of images of plants per time point for field 1. Note that the pre-defoliation orthophotos of October, 27th and October, 29th do not overlap the reference plots due to the low quality of the corresponding UAV images. Here, the reference plants were not defoliated, thus, the orthophotos of defoliation flights are used to extract images of these days to acquire a reference time series. For field 2, all local coordinates are given for all acquisition dates, which allows the user to extract complete image time series. Here, the data are divided into training, validation, and testing set for both fields. In addition, the plants in each plot are presented in each set. The visual distribution for both fields is shown in Fig. 5 in the Appendix.

3.2.4 Time Series for Defoliated Plant Data (GrowliFlowerD)

For field 1, the GrowliFlowerD subset contains a total of 130 plant IDs and coordinates for defoliated plants (30 for October, 27th and 100 for October, 29th). For field 2, the subset contains a total of 717 plant IDs and coordinates for defoliated plants. The coordinates allow the user to extract the time series of defoliated

Table 3.3: Number of defoliated plants per acquisition date for field 2 (2021).

Date	Aug 19	Aug 23	Aug 25	Aug 30	Sept 3	Sept 8
# Images	110	115	251	116	71	54

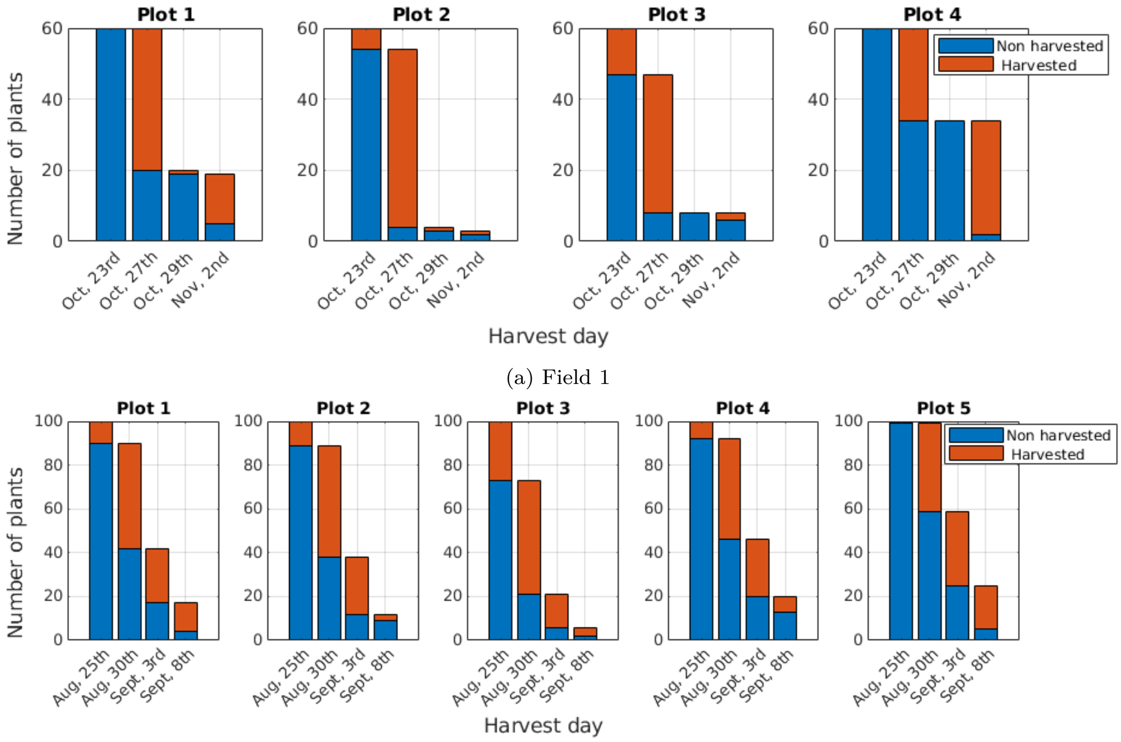
plants. Tab. 3.3 presents an overview of how many plants were defoliated on different acquisition days. In addition to the time-series data, pairs of pre- and post-defoliation images are provided in the subset. The data are divided into training, validation, and testing sets for both fields, and each defoliation day is presented in each set. The visual distribution of both fields is shown in Fig. 6 in the Appendix.

3.2.5 In-situ Data

Three CSV files are made publicly available, i.e., one for each field, and these files contain the plant ID and the measurements described in Sec. 2.3 for each data acquisition day. The measured values correlate with the images in the GrowliFlowerR subset. Fig. 3.6 shows the distribution of the number of harvested plants in the reference plots per acquisition date for fields 1, 2, and 3. A further subdivision into `Ready` and `Not-ready` plants per harvest day in training, validation, and test set is listed in Tab. 3.4. A similar distribution is given for each subset for the different HDs. The table shows that for HD$_1$, more data represent `Not-ready` for harvest plants. This is explained by the fact that as the plants grow, the proportion of harvest-ready plants increases, and thus the proportion of `Ready` for HD$_2$ - HD$_4$ increases and the proportion of `Not-ready` for harvest decreases.

Table 3.4: Comparison of `Ready` (1) and `Not-ready` (0) plants split into training, validation, and test set for each harvest day (HD), indicated by lowered digits.

	HD$_{1(0)}$	**HD**$_{1(1)}$	**HD**$_{2(0)}$	**HD**$_{2(1)}$	**HD**$_{3(0)}$	**HD**$_{3(1)}$	**HD**$_{4(0)}$	**HD**$_{4(1)}$	$\sum_{(0)}$	$\sum_{(1)}$
Train	261	37	117	144	32	85	0	32	410	298
Train	88%	12%	44%	56%	27%	73%	0%	100%	58%	42%
Val	78	8	25	53	7	18	0	7	110	86
Val	91%	9%	32%	68%	28%	72%	0%	100%	56%	44%
Test	72	14	28	44	8	20	0	8	108	86
Test	84%	16%	39%	61%	29%	71%	0%	100%	56%	44%



(a) Field 1

(b) Field 2

Figure 3.6: Overview of harvested and non-harvested plants per reference plot per day. Figure source: Kierdorf et al. [7].

3.3 Challenges in the Data

The datasets present certain challenges that must be considered during analysis. The theoretical size criteria specified by our project farmer is about 14 cm. However, practical realization of this criterion varies, influenced by various factors including curd quality, meteorological conditions, economic considerations, and field labor availability. These factors, coupled with different plant growth stages [212]–[214] lead to multiple harvest-runs on the same field throughout the harvest period resulting in multiple weeks of harvest [215]. There exists a temporal gap of 2-4 days between two successive harvest days, whereby the plants may be ready for harvest 1-3 days before the next harvest-run, but will not be considered until the next planned harvest-run. Regarding the reference data for the harvest day, it should be noted that during a harvest round, not necessarily the entire field is checked for harvest-ready heads. This depends on the field size and the remaining working time. A rough overview is obtained, and if no harvest-ready heads are found in large areas, the rest of the field may not be checked. As a result, some plants that might have been ready for harvest could be missed due to the lack of field monitoring. During data collection and processing, we assume that plants with harvested curds are deemed ready for harvest. In practice, however, this may also have been due to misjudgment by the field worker or inadvertent damage to plants during the harvesting process.

For the image data, the varying weather conditions for each flight result in differences in lighting and image quality over time. This variation occurs not only within a monitoring period of a field but also across different fields and years. The data quality and quantity vary in terms of lighting, data availability, blurriness, and other factors. Different varieties were planted in various fields, leading to differences in plant appearance. The varieties exhibit varying degrees of self-coverage, which in some cases necessitates intervention by field workers to cover the cauliflower curd.

Chapter 4

Baseline for Instance Segmentation Application

4.1 Experimental Setup

We describe two possible applications of the proposed dataset by creating baselines using the labeled GrowliFlowerL subset and the Mask R-CNN [105], a state-of-the-art instance segmentation method. We use the pytorch implementation available at https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html.

We consider plant instance and leaf instance segmentation tasks. Thus, we use the mask and bounding boxes derived from the plant instance mask as the target for the first baseline. The mask and bounding box derived from the leaf instance mask are used as the target for the second baseline. For the leaf instance segmentation baseline, the given void instances are used as the background because only leaves that do not belong to void plants are labeled. Note that the estimation of semantic masks for individual instances enables the derivation of phenotypic traits. Here, we applied a random horizontal flipping data augmentation technique with a probability of 0.5.

We trained the Mask R-CNN on a computer with an Intel Core i7-6850K 3.60 GHz processor and a GeForce GTX 1080Ti GPU with 11 GB RAM. The network was pretrained on the COCO dataset [186], and training was performed over 100 epochs with a learning rate of 0.001 and batch size of 2. We used an SGD optimizer, and ResNet-50 was used as the backbone network.

4.2 Evaluation Metrics

We compute precision, recall, and F1 score relative to the single object class cauliflower plant and calculate the scores for the IoU thresholds $t_{\mathrm{IoU}} = 0.50$ and $t_{\mathrm{IoU}} = 0.75$. In addition, we determine the average precision (AP), average recall

(AR), and average F1 (AF1) scores over all IoUs in the interval $0.50 - 0.95$ with a step size 0.05 as for the COCO benchmark. This is indicated by $(\cdot)@0.5-0.95$. For the leaf instance segmentation baseline, we reduce the evaluation on recall, as we do not want to penalize predictions on void pixels. The consequence of penalizing predictions on void pixels would be to penalize the model in identifying leaves that were simply not labeled as such.

4.3 Experimental Evaluation

We calculate the metrics relative to the detected bounding boxes and the segmented masks of the respective objects. The segmented masks provide information about the cumulative number of correctly classified pixels and, thus, the more accurate shape of the object. The bounding box enables derivation of the detection accuracy and thus, the localization of the object.

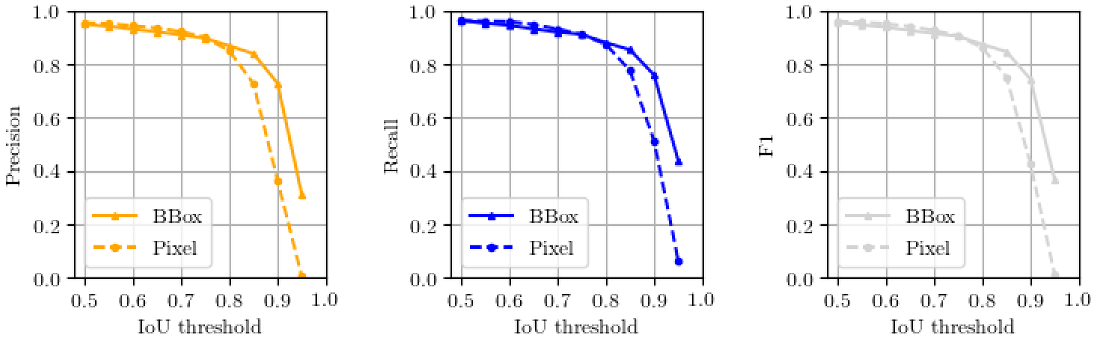
Tab. 4.1 summarizes the results for the plant instance segmentation task for the baseline method. As can be seen, 95% at IoU ≥ 0.5 are predicted correctly. In addition, precision at the bounding box and pixel levels are greater than 80% for all IoU thresholds ≤ 0.8 (Fig. 4.1a). At an IoU value of ≥ 0.85, precision decreased rapidly. This trend is also observed for both recall (Fig. 4.1b) and the F1 score (Fig. 4.1c). For higher IoU values, we found that prediction at the pixel level is less accurate than at the bounding box level because slight changes in segmentation generally result in more errors in the segmentation mask than in the bounding box. An overview of the results is given in Tab. 4.1.

We found that many objects and masks are estimated accurately (Fig. 4.3a). The results show all predictions with a confidence score greater than a threshold of 50%. Precise contours are estimated, and in the earlier development stages, the instances are well separated spatially. Note that the model does not predict the ground as an object in any case. In addition, smaller weeds that can be seen in some patches are also not considered objects, which is desirable because, in this way, we identify that the model distinguishes cauliflower from weeds. We found that inaccuracies occur with plants that lie at the edge of the image patches. In such cases, only small parts of the plant are visible, thus, the leaves are not adjacent to each other, as shown in Fig. 4.3b (top left and bottom left). We also found that errors occur in later developmental stages because the plants overlap (Fig. 4.3b bottom right), which represents a more challenging scenario than well-separated plants. In particular, for overlapping plants, it is even difficult for the human eye to assign leaves to individual instances. In addition, compared to the earlier stages of development, where no overlap occurs, fewer training images were available for the later stage of development. The small number of images means that less variability in the data is captured, making predictions on new unknown

Table 4.1: Plant instance segmentation results: precision, recall, and F1 score for predicted bounding boxes (BBox) and segmentation masks (pixel) for the class `plant`.

	Global metrics			Precision		Recall		F1	
Evaluation	**AP**	**AR**	**AF1**	**P@0.5**	**P@0.75**	**R@0.5**	**R@0.75**	**F1@0.5**	**F1@0.75**
BBox	0.917	0.933	0.843	0.952	0.899	0.965	0.913	0.958	0.906
Pixel	0.844	0.858	0.770	0.954	0.902	0.963	0.913	0.959	0.908

Abbreviations: AP, average precision; AR, average recall; AF1, average F1.



(a) Precision

(b) Recall

(c) F1

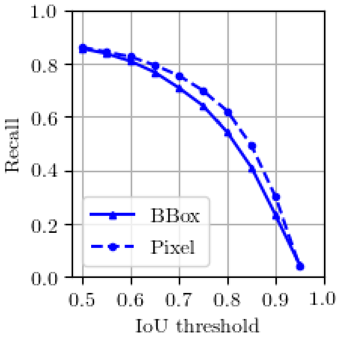
Figure 4.1: Representation of precision, recall, and F1 score for class cauliflower `plant`. The graphs show the evaluation at different IoU thresholds on the bounding box (BBox), thus object, (solid-line) and pixel (dashed-line) level. Figure source: Kierdorf et al. [7].

data more difficult.

Another distinctive feature involves plant objects from which leaves fall or plants that are impaired in their growth and thus decay. In such cases, it is difficult for the model to distinguish whether one or more plants are represented (Fig. 4.3b top right).

For the leaf instance segmentation task, which is a more difficult task compared to plant instance segmentation, we achieve a very good recall result of 74% at the bounding box level and 77% at the pixel level. The distinction between individual leaf instances is more complex than the distinction between plant instances. In addition, here, we assign the void labeled objects to the `BG` class for this baseline rather than the `leaf` class because individual void plants can contain several leaves, however, such leaves were not labeled individually. Note that the calculated values for recall are similar to both the pixel and bounding box levels.

We can find explanations for the recall values in the visual consideration of the results, even though these results show predictions with a confidence score greater than a threshold of 50%. By defining void instances as the background, the model is challenged to predict the leaves belonging to void instances not as leaf objects, as shown in Fig. 4.3d (top left and bottom left). It is difficult for the model to distinguish whether plants at the edge of the patches are void instances or leaf



(a)

	Global metrics	Recall	
Evaluation	**AR**	**R@0.5**	**R@0.75**
BBox	0.735	0.857	0.642
Pixel	0.772	0.862	0.700

(b)

Figure 4.2: Recall results for leaf instance segmentation task. Graph (a) shows the evaluation at different IoU thresholds on the bounding box (BBox), i.e. object (solid-line), and pixel (dashed-line) level. (b) shows the respective average recall (AR), R@0.5, and R@0.75 values. Figure source: Kierdorf et al. [7].

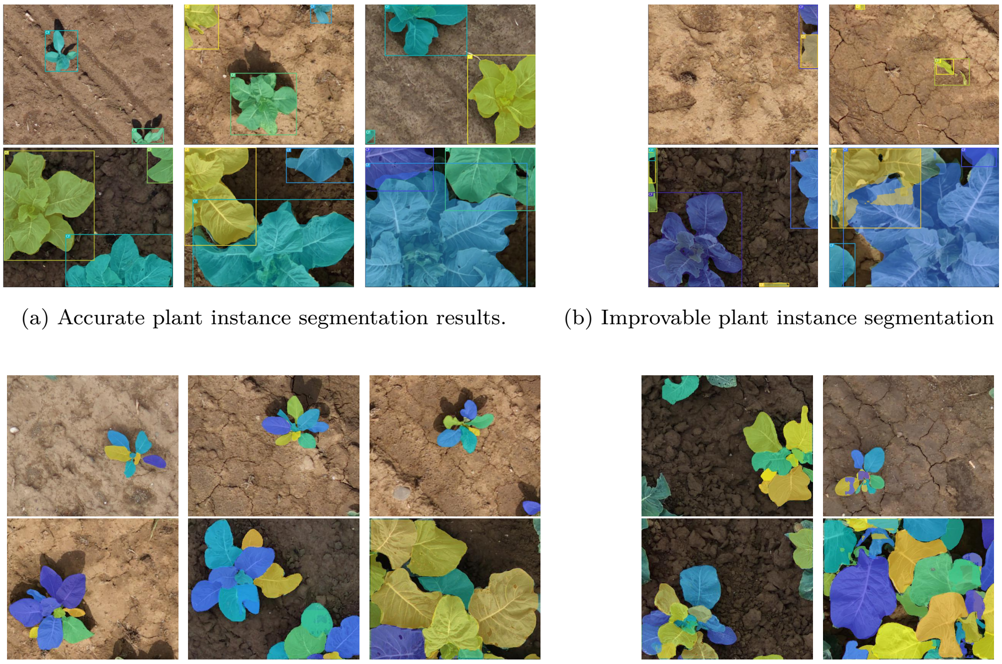
instances. Thus, either leaves are predicted that are not in fact present in the target (representing low precision) or no leaves are predicted even if they are present in the target (representing low recall). For plants that are completely visible in the patch, the model demonstrates better prediction performance. Another source of error is the prediction of several instances on a single leaf, as shown in Fig. 4.3d (top right and bottom right) because the model is required to learn various features, e.g., leaf structure and size. After all, such features play a crucial role in distinguishing different leaves.

To sum up, we observe that our instance segmentations, plant instance as well as leaf instance, perform and can be used for different growth stages of the cauliflower plants.

4.4 Reflection and Discussion

The results of this research provide insight into the acquisition of image time series under field conditions. We observe that the flight altitude of the UAV must be adjusted depending on the characteristics and the height of the cultivated plants in order to capture images with high quality in terms of resolution and gapless spatial data. However, to obtain accurate image data, enough keypoints must be distributed in the field. Our work shows that GCPs are suitable as keypoints because they help to align and georeference the orthophotos more accurately. For simplified use of the data for ML approaches, the data should have similar characteristics as exposure over time. For this purpose, data must be recorded under consistent weather conditions. However, we note that combining consistent conditions and similar interval lengths between acquisition days is a challenging task.

While previous research, such as the work of Bender et al. [81], focuses on



(a) Accurate plant instance segmentation results.

(b) Improvable plant instance segmentation results.

(c) Accurate leaf instance segmentation results.

(d) Improvable leaf instance segmentation results.

Figure 4.3: Plant and leaf instance segmentation results. The different colors indicate the different instances. In the visualization of the leaf instance segmentation results, we concentrate on the visualization of the masks for clarity and omit the bounding boxes. The examples shown in (a) and (c) show accurate results and those in (b) and (d) show improvable results. Figure source: Kierdorf et al. [7].

collecting data from many different data sources (e.g., imagery, climate, and soil data), monitoring a large number of different instances that are specifically needed for training DL methods has not been considered sufficiently. In our work, we provide a large image number of different instances but we lack additional data such as soil and climate data. Climate and soil characteristics are important factors to determine or predict plant growth. ML methods only learn features that are present in the training data. Thus, if external factors such as climate and soil change, the growth of plants is influenced as well. The lack of this information can cause ML models to be prone to errors in their results when applied to new data. Therefore, we suggest improving the data acquisition by capturing additional soil and climate data. Another suggestion for improvement of data acquisition is the field design. To avoid systematic effects in the data, reference plots and other selected areas like defoliation plots should be distributed equally within the field because it is difficult at the beginning of data collection to predict how much which plants (crops and weeds) will grow and how the location will affect growth and thus, the acquired data.

Regarding our baseline experiment considering instance segmentation, we observe that the application shows satisfactory results on our data. To improve the results, future studies could integrate prior knowledge about the shape and structure of plants and leaves. Weyler et al. [216] show an example of how to improve the results for plant and leaf segmentation by developing a combined approach of neural networks and clustering to simultaneously determine leaf and plant instances. However, to the best of our knowledge, the approach is not used for plants as large and highly overlapping as in our provided later developmental stages (see Fig. 4.3c and Fig. 4.3d).

Another way to improve the result of instance segmentation is to vary the field of view per image. In further experiments, we observe that when applying to a modified field of view, plants are segmented with only a few errors. For these experiments, we reduce the threshold by up to 20% depending on the extension of the field of view. The masks and bounding boxes of the predictions match the plants. However, the size of the objects differs from the training data due to the change in scale. This causes the model to have less confidence in its predictions, even if they are correct. Even with a changed image size, cauliflower plants can be easily distinguished from weeds and each other. As it brings greater variability to the data, adding images with a larger field of view to the training set could lead to further improvements in the results. However, this would require labeling more data.

We recommend our dataset for further methodological developments or as an evaluation dataset for existing approaches as used in our cooperative work of Günder et al. [15].

Chapter 5

Conclusion and Outlook

In this article, we have introduced the GrowliFlower dataset, a georeferenced, image-based UAV time-series dataset of two monitored cauliflower fields during their entire growth period. The proposed dataset was described, and we discussed the data collection process, which may be helpful for other similar data collection procedures. The proposed dataset comprises weekly RGB and multispectral UAV orthophotos and image time series of individual plants reflecting weekly plant growth. In a subset of the proposed dataset, in-situ reference measurements, e.g., plant size, are also available, and another subset provides pre- and post-defoliation images to demonstrate the relationship between the interior and exterior of the cauliflower plant. The proposed dataset also contains annotations with segmented plant and leaf instances, as well as annotations on stems. The data are available at *http://rs.ipb.uni-bonn.de/data/*. The proposed dataset is intended to promote using and evaluating ML methods and foster close collaboration between disciplines, e.g., agricultural sciences, remote sensing, and machine learning. We have also presented baseline results of two applications of the proposed dataset using the Mask R-CNN model, i.e., plant instance segmentation and leaf instance segmentation tasks. In addition, we expect that the findings and descriptions presented in this paper will help realize effective data collection processes that are transferred to other areas. The steadily increasing citations and downloads of the dataset indicate that interest in the GrowliFlower dataset is continuously growing.

Part II

Single Image Time Point Analysis

Introduction

In this part of the thesis, we address the task of harvest-readiness prediction of single cauliflower plants which can be challenging due to the cauliflower head being covered by its canopy. Here, we specifically examine individual growth stages shortly before harvest without integrating information about plant development or other factors. While deep learning enables automated harvest-readiness prediction, errors can occur due to field variability and limited training data. To tackle these errors, we aim to derive a reliability score for the model's output that can be used to support the farmers in their decision-making process.

To reach our goal, we use saliency mapping to identify image regions that have distinctive characteristics important for the model decision [217], [218]. We extend the clustering approach of saliency maps by Lapushkin et al. [195] and combine the maps with knowledge about our application domain and the image properties to derive reliability scores of the model's output. Our work differs from related work in that we propose a framework for deriving a reliability score for classification predictions that operates post-hoc during inference time without human interaction. Thus, the system can be applied to already trained models without changing the model architecture and without the need for re-training.

The main contributions of this part are:

- two image-based classification models for assessing the harvest-readiness of cauliflower: A binary classification model that achieves an overall accuracy and balanced class accuracy of 79% each, and a multi-class classification model that reaches an overall accuracy of 63% and a balanced class accuracy of 48%.
- a versatile post-hoc approach to derive intuitive reliability scores without time-consuming human interaction;
- a use case where the reliability scores are used to improve harvest-readiness predictions on the GrowliFlower dataset by 16.84% to an overall accuracy of 93.88% and by 16.30% to an balanced class accuracy of 93.91%.

The majority of this part is based on our published paper by Kierdorf et al. [8]. Further contents were taken from our paper submitted by Emam et al. [10]. Additions have been made regarding the forecasting time of classification, the models used, and the framework steps, leading to changes in the results compared to Kierdorf et al. [8].

Chapter 1

Scenarios

In the following, we present various scenarios that describe different approaches to achieving the objective of predicting cauliflower harvest-readiness. We elucidate which of these scenarios are applicable in this study and which are not. For each scenario, we illustrate a timeline depicting data acquisition and tactical management decisions like harvest, similar to the example shown in Fig. 1.1. All rows in the figures depict the same timeline. Each row describes the occurrences of data acquisition, processing, analysis, and tactical management decisions within a certain timeline for a harvest-time-point T, starting with T_1 and extending over H harvest days. A gray circle denotes a time point in the time series, while a yellow circle represents the day of data acquisition, processing, and analysis. In contrast, a blue circle signifies the day tactical management decisions are made.

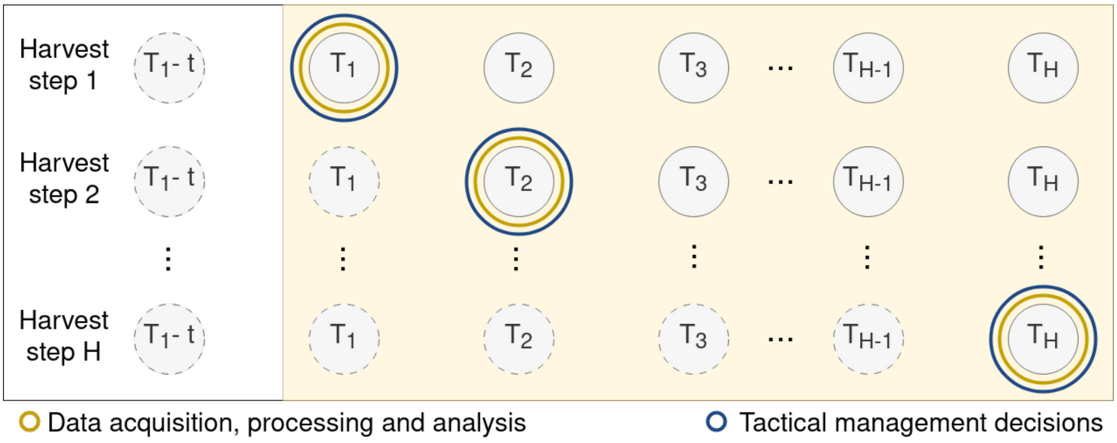

Harvest step 1
Harvest step 2
Harvest step H
T1- t
T1
T2
T3
...
TH-1
TH
○ Data acquisition, processing and analysis
○ Tactical management decisions

Figure 1.1: Scenario 1 visualizes the interaction between data collection, processing, and analysis by applying tactical management decisions for the binary classification of harvest-readiness based on single time point data. In scenario 1, the actions are carried out on the same day.

Scenario 1

The first scenario illustrated in Fig. 1.1 closely resembles the manual harvest process, aiming to ascertain the harvest-readiness of a plant on a given day T_h. Therefore, data acquisition, processing, and analysis occur on the same day to inform tactical management decisions in the field. However, due to the time constraints associated with data collection, processing, and analysis, this scenario is not practically feasible. Farmers typically plan ahead, allocating field workers to specific crops and fields and commencing work early in the day. Consequently, we exclude this scenario from further consideration in this thesis.

Scenario 2

In the second scenario, depicted in Fig. 1.2, data acquisition, processing, and analysis are conducted t days before harvest to facilitate proactive planning for farmers and potential resource reallocation. In this scenario, the investigation revolves around whether a plant will be considered ready for harvest in t days from day T_h. This includes the consideration of a binary problem. Furthermore, the forecasting time t is examined to ensure accurate results. Following harvest on day T_h, the next drone flight is conducted to again analyze harvest-readiness of plants T_{h+1}, and so on. This scenario's effort is logistically feasible and enables the farmer to effectively plan and allocate resources. By determining the harvestable quantity, the farmer can exert some influence on the market. However, within this scenario, no estimation is available for planning the entire harvest phase. During the harvest period, a drone flight must be conducted after each harvest.

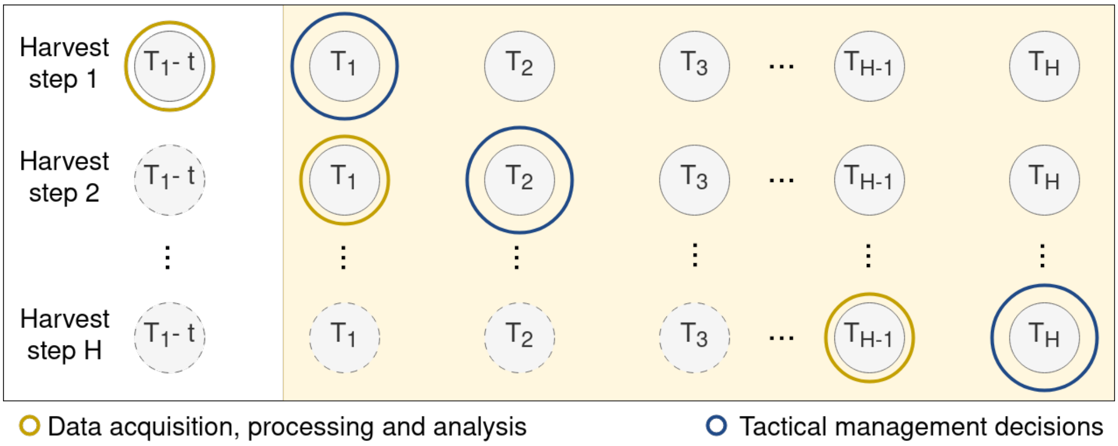


Figure 1.2: Scenario 2 visualizes the interaction between data collection, processing, and analysis by applying tactical management decisions for the binary classification of harvest-readiness based on single time point data. In scenario 2, the actions are carried out on different days with a lead time t days.

Scenario 3

In the third scenario, we omit drone flights during the harvest and investigate the harvest time point T_h for a plant. This scenario is depicted in Fig. 1.3. We approach this scenario as a multi-class classification problem. Similarly to scenario 2, it is interesting to determine the duration t between data acquisition and harvest time T_1 to achieve the most accurate results. Compared to scenario 2, this scenario offers the same advantages. Additionally, the workload of data acquisition, processing, and analysis is significantly reduced, as it only needs to be conducted on a single day $T_1 - t$. Tactical management decisions can be planned early, making this scenario applicable in practical implementation.

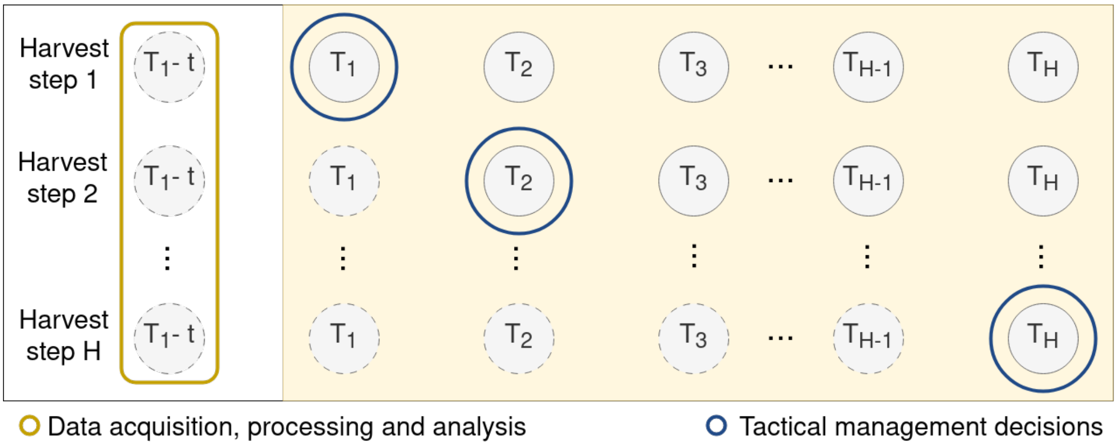


Figure 1.3: Scenario 3 visualizes the interaction between data collection, processing, and analysis by applying tactical management decisions for the multi-class classification of harvest-readiness based on single time point data. In scenario 3, data collection, processing, and evaluation are only carried out once, with a lead time of t days before the first harvest day, which is T_1.

Chapter 2

Binary Harvest-readiness Classification

Accurately classifying cauliflower's harvest-readiness state is pivotal for optimizing agricultural operations and ensuring the timely and efficient harvesting of crops. Conventional methods often rely on manual inspection, which can be labor-intensive, subjective, and prone to inconsistencies. By utilizing the capabilities of neural network models such as ResNet-18 and ViT-B/16, we attempt to automate the classification of cauliflower harvest-readiness.

In this study, we embark on a comparative analysis of these neural network architectures to ascertain their efficiency in accurately classifying cauliflower harvest-readiness. Despite their different design architectures, both ResNet-18 and ViT-B/16 are known for their capabilities in image classification tasks. Two primary expectations drive our experiment. Firstly, we anticipate comparable accuracies between ResNet-18 and ViT-B/16. Furthermore, we hypothesize that models utilizing data from three days before harvest will yield higher accuracies compared to those using data from six days before harvest. This expectation is grounded in the understanding that cauliflower growth is heavily influenced by weather conditions, with the closer proximity to harvest providing more relevant and reliable data for prediction. By exploring these hypotheses, we aim to identify the most effective approach for cauliflower harvest-readiness prediction based on scenario 2.

2.1 Experimental Setup

We use two image sets of the GrowliFlowerR dataset of field 2 [7]. The first consists of images of the dates 2021-08-23, 2021-08-25, 2021-08-30, and 2021-09-03 with given information about harvest-readiness within the next three days. The second set consists of images around 6 days before harvest, 2021-08-19, 2021-08-23, 2021-08-25, and 2021-08-30. Example images are shown in Fig. 2.1. Three days

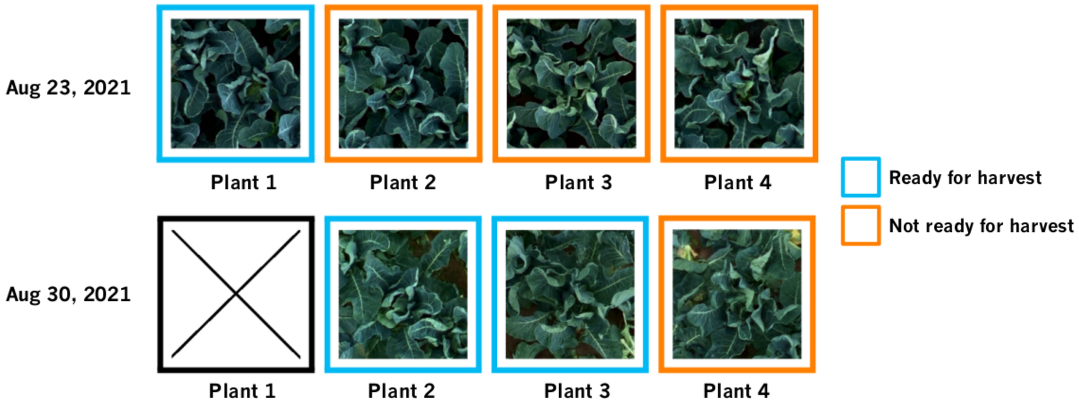


Figure 2.1: Example of `Ready` and `Not-ready` for harvest plants. A column represents the same plant at different times, depicted by the rows. The difficulty of accurately classifying into `Ready` and `Not-ready` for harvest becomes evident in this example due to the similar visual characteristics of the classes simultaneously.

is a compromise between harvest-readiness prediction accuracy and practicability of data acquisition. Six days lead time extends the previously mentioned points by a longer forecasting time for improved practicability in the field. It is important to note that the lead time between image acquisition and harvest includes a buffer, which means that it does not strictly predict harvest-readiness for points in time precisely 3 or 6 days ahead. We divide the data into the classes `Ready` and `Not-ready` for harvest. The plants representing both classes show high similarity between the same day of acquisition and between different days. The size of their curds determine the ripeness, however, in most images, the canopy covers the curd. The plant's stem is centered within the image, but depending on the plant's growth, the center of the cauliflower curd can vary up to 20 cm from the stem.

We use the training, validation, and test splits as described in [7]. If the plant shown in an image is already harvested, we exclude the image from the dataset. This results in preliminary training sets of 541 images, validation sets of 196 images, and test sets of 194 images. We apply standard augmentations like flipping and rotation on the training data. For images of class `Not-ready`, we apply augmentations 50% more often than for images of class `Ready` to get a more balanced data distribution. After data augmentation, the training sets contain 6224 images, 2432 of class `Not-ready`, and 3792 images of class `Ready`.

For this experiment, we use the ResNet-18 and ViT-B/16 for single-input images. For each architecture, we train one model with a forecasting time of about three days and one model with a forecasting time of about six days. The training for each model consists of at least 50 epochs and stops if validation accuracy does not increase significantly over 10 epochs. We use an Adam optimizer with a weight decay of 0.0001. The learning rate starts at 0.0001 and is reduced with a learning rate scheduler with a step size of 5 and factor γ of 0.1. We utilize several evaluation metrics, including overall accuracy (oaAcc), balanced class accuracy (bcAcc), recall, and precision, whereby recall and precision are calculated for class `Ready`. Furthermore, we examine the oaAcc per harvest day.

2.2 Experimental Evaluation

Upon comparing the evaluation metrics presented in Tab. 2.1, it becomes evident that both ViT-B/16 models exhibit slightly higher accuracies in terms of overall accuracy and balanced class accuracy compared to the ResNet models. The observed differences range up to 2%. As informed in the basic methodology chapter, this similarity may stem from the dataset's size. The limited scope of data and constrained variability therein may result in the absence of inductive biases. Another plausible explanation for non-significant differences could be attributed to the interplay between the task and the model architecture. Given the classification of a state rather than an object, spatial information or the receptive field within a CNN likely proves more advantageous for classification compared to the similarity of patches within a ViT, thus reducing the advantages of ViT and resulting in similar accuracies.

For all four models, we achieve a higher recall compared to precision. The high recall values indicate that many `Ready` for harvest plants were correctly classified. In practical field applications, this could potentially reduce the need for field workers to manually inspect the entire field for harvest-ready plants. Instead, they would only need to cross-check the plants classified as `Ready` for harvest to identify falsely classified plants for selection.

The accuracies for the two forecasting times exhibit remarkable similarity despite the expectation that data acquired closer to harvest would yield enhanced accuracies due to the significant influence of weather on cauliflower plant growth. To further investigate this, we examine the oaAcc across individual harvest days. Our analysis reveals that the models accurately capture the underlying data distribution, as evidenced by comparison with Tab. 3.4. Notably, each harvest day include a class distribution imbalance, which could not be improved by applying data augmentation. While the augmentations effectively made the class distribution more similar, the discrepancies in the within-day distribution were not specifically considered. Consequently, the underlying data exhibit disparate appearance characteristics across distinct harvest days due to varying weather conditions during data acquisition periods. In collaborative work with Penzel et al. [11], our investigation centered on the trained ResNet$_3$ model, seeking to unravel causative relationships between certain features and prediction outcomes. Among

Table 2.1: Harvest-readiness classification accuracies for single time points. Comparison of ResNet-18 and ViT-B/16 model for forecasting harvest-readiness in approximately 3 or 6 days. Accuracies are displayed in %.

(a) Validation set

Model	oaAcc	bcAcc	Recall	Precision	$\text{oaAcc}_{\text{HD}_1}$	$\text{oaAcc}_{\text{HD}_2}$	$\text{oaAcc}_{\text{HD}_3}$	$\text{oaAcc}_{\text{HD}_4}$
ResNet-18_3	77.04	77.52	81.46	70.70	84.88	66.67	76.00	100.00
ResNet-18_6	79.08	80.35	90.70	70.27	84.84	73.08	72.00	100.00
ViT-B/16_3	78.06	78.93	80.23	72.63	89.53	67.95	64.00	100.00
ViT-B/16_6	79.59	80.68	89.53	71.30	89.53	70.51	68.00	100.00

(b) Test set

Model	oaAcc	bcAcc	Recall	Precision	$\text{oaAcc}_{\text{HD}_1}$	$\text{oaAcc}_{\text{HD}_2}$	$\text{oaAcc}_{\text{HD}_3}$	$\text{oaAcc}_{\text{HD}_4}$
ResNet-18_3	72.16	72.75	77.91	65.69	83.72	55.56	75.00	87.50
ResNet-18_6	74.22	74.72	79.07	68.00	80.23	66.67	71.43	87.50
ViT-B/16_3	74.74	75.54	71.43	75.58	83.72	66.67	67.86	100.00
ViT-B/16_6	73.71	74.38	80.23	66.99	83.72	63.88	67.86	75.00

our findings, we discovered that average brightness exerts a discernible influence on prediction results. We assume that these observations extend to other models under consideration.

2.3 Conclusion

We classify the harvest-readiness of cauliflower plants using a convolutional neural network-based ResNet-18 classification model and an attention-based Vision Transformer model, considering different forecasting times for harvest-readiness.

The comparison between the accuracies achieved by the two models shows results without significant differences. We attribute this to the amount of data and the fundamental structures of both models. Similarly, no significant difference is observed between the different forecasting times. We confirm this by noting that the models take features such as the lighting conditions at various capture times into account for their decisions rather than just the visual appearance of the plant.

Chapter 3

Reliability Scores from Saliency Map Clusters for Classification Improvement

Farmers and others outside the ML community are often skeptical when ML models are involved, especially regarding economic profit. Thus, it is important to find an explanation for the model's decision to get a better understanding. Therefore, interpretation tools creating saliency maps help. They provide a visual explanation of the model's decision. The maps help to understand and improve models but also help to derive reliability statements.

In this chapter, we propose a framework for deriving a reliability score for classification predictions that operates post-hoc during inference time without human interaction. Thus, the system can be applied to already trained models without changing the model architecture and without the need for re-training. This chapter is built on scenario 2.

3.1 Experimental Framework

We solve the task of estimating the harvest-readiness of single cauliflower plants with deep learning-based image classification and combine it with an estimation of the reliability of the classification through group assignments of saliency maps. Fig. 3.1 shows an overview of the five-step framework.

1. *Classification:* In the first step, images are classified into the classes `Ready` and `Not-ready` for harvest within three days. We use a ResNet-18 network, however, the framework is flexible regarding the classifier.

2. *Saliency Mapping:* In the second step, we compute saliency maps for validation and test data post-hoc using the learned classifier. We consider

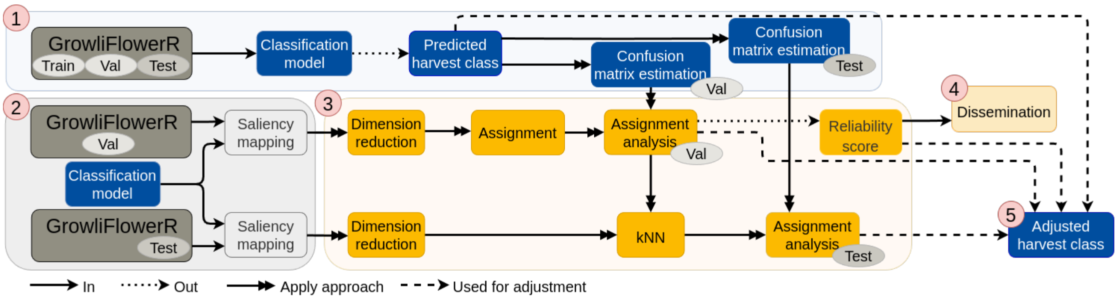


Figure 3.1: Our framework for deriving reliability scores. The different numbers represent (1) the classification step, (2) the saliency mapping step, and (3) the assignment step of saliency maps with the assignment of reliability to the groups by relating the confidence scores of the model to the corresponding saliency maps. (4) represents the dissemination to the farmer of how reliable the model is, while (5) represents the adjustment step, where the predictions of (1) are improved by using the reliability score of (3). The figure is adapted from Kierdorf et al. [8] and slightly modified.

Grad-CAM, OSM, and LIME.

3. *Assignment:* We employ Spectral Clustering, Expectation-Maximization, and Kernel Density Estimation to identify groups of saliency maps computed on validation data and derive reliability scores. For clustering techniques, the mean saliency map per cluster, denoted as a prototype, is further analyzed. Test data can be assigned a reliability score by assigning its saliency map to the nearest group.

4. *Dissemination:* The reliability score is intuitively usable due to its value range between 0 and 1 and is communicated to the user together with the classification result.

5. *Adjustment:* In our use case, the classification results are adjusted based on the group assignment of the saliency maps to determine the final predicted classes. Using a clustering approach, the decision depends on the summed percentage of false positives (FP) and false negatives (FN) per cluster. For Kernel Density Estimation, the decision depends on the confidence interval of log-likelihood values derived from true predictions. The evaluation of the classification step provides the assignments to true and false predictions.

The framework does not require human interaction and can be applied to different models. However, human interaction is possible to improve the classification results and reliability measures further by analyzing and evaluating the human-understandable groups of saliency maps.

Classification

In previous experiments detailed in Chap. 2 of this part, we investigate various models for binary harvest-readiness classification. Subsequently, for further explorations within this experiment, we employ the model ResNet-18_3. We assume that our investigations are equally applicable to the other models and would yield similar results.

Saliency Mapping

Saliency maps aim to explain a model's decision by identifying important regions in the image. In our case, saliency maps highlight which image regions are important for predicting the classes `Ready` and `Not-ready` for harvest, allowing conclusions about the reliability using the prior knowledge that the center of the image is important for the decision and the background should not play a role in the harvest-readiness estimation. We consider three well-established local approaches as baseline approaches for saliency mapping, namely a gradient-based approach, Grad-CAM, and two permutation-based approaches, OSM and LIME, where LIME differ in that it uses surrogate modeling, as our focus is not on the used methods.

Parameters need to be set for the various approaches. As the resulting Grad-CAM map depends on the employed layer, we follow the suggestions of Selveraju et al. [121] to use the last convolutional layer as it highlights object-level regions in the image, which are also easier to interpret. Grad-CAM provides information about the class of interest but no information about other classes. For the second approach OSM, parameters such as stride s and patchsize p must be specified. We select $s = 2$ to maintain a high resolution and $p = 11$ as the patchsize. With the chosen patchsize, it is feasible to perturb roughly a quarter of the cauliflower curds simultaneously. For LIME, we use a least squares linear regression model.

Group assignment

We adopt the idea proposed by Lapuschkin et al. [195], employing unsupervised techniques to group the resultant saliency maps, which provides a better understanding of the model decision by taking into account image features other than RGB. This helps to reduce direct influences such as exposure, which, as mentioned in Chap. 2, can have a negative impact on the classification results and may also have a negative result on the group assignment results. Our study conducts a comparative analysis of three prevalent unsupervised techniques: Spectral Clustering (SC), Expectation-Maximization (EM), and Kernel Density Estimation (KDE). To this end, we leverage the validation set, comprising V samples

$\{\mathbf{x}_1^{\mathrm{IT}}, \ldots, \mathbf{x}_V^{\mathrm{IT}}\}$, which represent the saliency maps for each validation sample $\mathbf{x}_v$. Before we apply the unsupervised techniques, we perform principal component analysis (PCA) on the vectorized saliency maps, reducing the dimensionality from 65536 to 50. This dimensionality reduction is motivated by the goal of retaining 95% of the variance, given the absence of significant differences in successive eigenvalues.

As a first technique, we apply SC to the validation set, as described by Lapuschkin et al. [195], then assigning the nearest cluster IDs to the test data using kNN with $k = 5$. For this approach, we set the number of clusters $q = 9$ to ensure representativeness and generalizability across diverse datasets.

Secondly, we apply EM to the validation set and determine the closest cluster assignment by computing the probability of a data point belonging to the Gaussian distributions representing the clusters. This probabilistic assignment contrasts with traditional k-Means clustering, allowing data points to be associated with one or more Gaussian distributions with varying probabilities. For our analyses, we exclusively consider assignments with maximal probability.

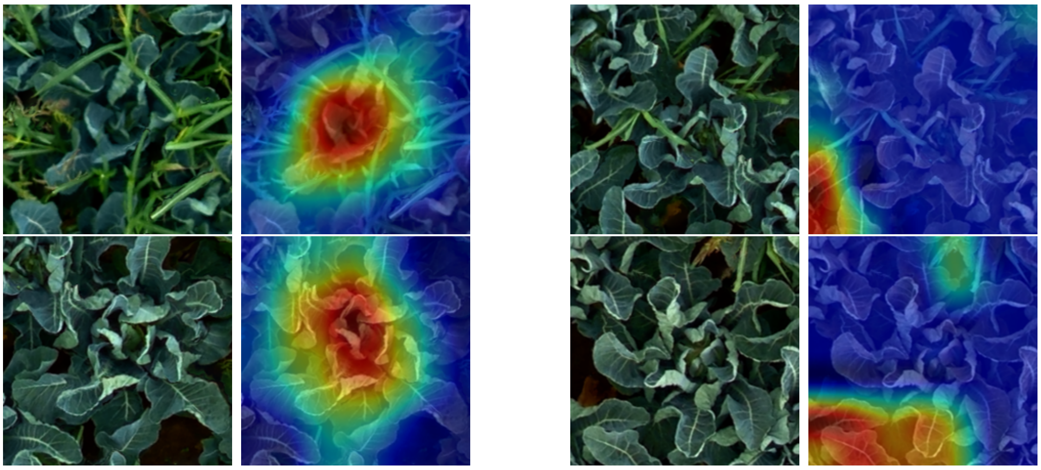
Thirdly, we apply KDE to the true predictions of the validation set, depending on a selected bandwidth h that maximizes the log-likelihood values of the validation samples. We use the calculated PDF as the basis for our comparison. We define samples inside the confidence interval of the PDF as the confidence-group and samples outside the confidence interval as the non-confidence-group. For our analyses, we determine the log-likelihood values of the false predicted samples of the validation data, as well as the log-likelihood values of the entire test data and assign the samples to the corresponding group.

Dissemination

In the dissemination step, we leverage prior knowledge to draw conclusions about reliability. This includes the understanding derived from data processing that the cauliflower curds are positioned at the center of the images. At the image borders, neighboring plants or soil are visible, which are not intended to influence the classification outcome. Examples of how such maps could look like are shown in Fig. 3.2.

Evaluation Metric

We differentiate the evaluation of the adjustment step for SC and EM on the one hand and KDE on the other hand. To evaluate the adjustment step for SC and EM, the summed percentage of FP and FN is considered in the calculated clusters q. We define $r_q = 1 - (\mathrm{FP}_q + \mathrm{FN}_q)$ as reliability score. The higher the reliability score, the more reliable a prediction is in a specific cluster. If r_q falls below a



(a) Characteristics that correlate with expert knowledge.

(b) Characteristics that do not correlate with expert knowledge.

Figure 3.2: Examples of saliency maps showing characteristics that correlate with expert knowledge in (a) and do not correlate with expert knowledge in (b). For better spatial understanding, the maps are overlaid with the RGB input image.

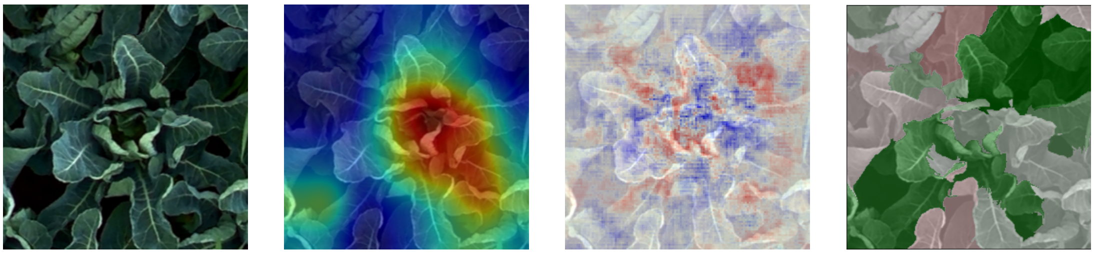
threshold t in cluster q of the validation set, we swap the predicted class for all samples in cluster q and update the confusion matrix. We choose $t = 25\%$ for our experiments. Threshold t is variable and selectable based on data and trained model and should be chosen to significantly improve the validation set's accuracy. Based on the updated confusion matrix, we adjust the overall and average class accuracy. We store the identified clusters for swapping and apply the same to the test data, followed by updating the test confusion matrix and accuracies.

To evaluate the adjustment step for KDE, we compute the 95% confidence interval of the PDF. If the log-likelihood value of a sample falls within the non-confidence-group, we set the reliability score r to 0 and swap the predicted class for the specific sample, and update the confusion matrix and accuracies. If the log-likelihood value falls within the confidence-group, we set r to 1.

3.2 Experimental Evaluation

3.2.1 General Discussion

Our experiments find that clusters or probability densities do not correlate with harvest-readiness classes. This is expected in binary decision-making, where both classes may end up in the same cluster or have similar data distributions since they ideally use the same features. Instead, we focus on whether data within a cluster or distribution are correctly classified, which allows conclusions to be drawn about the reliability of the result. We use the confusion matrix for analysis. To assist the farmer in making harvesting decisions, we exploit the fact that the saliency



(a) RGB (b) Grad-Cam (c) OSM (d) LIME

Figure 3.3: Resulting saliency maps or the used approaches (b) Grad-CAM, (c) OSM, and (d) LIME for a RGB input image (a) which is visualized in the maps' background. Figure source: Kierdorf et al. [8].

maps of plant images whose classification result is primarily on the main diagonal of the confusion matrix (TP or TN), and maps that are associated with incorrect classification results (FP or FN) tend to end up in separate clusters. Furthermore, the incorrect classifications tend to fall outside the 95% confidence interval of the log-likelihood value distribution.

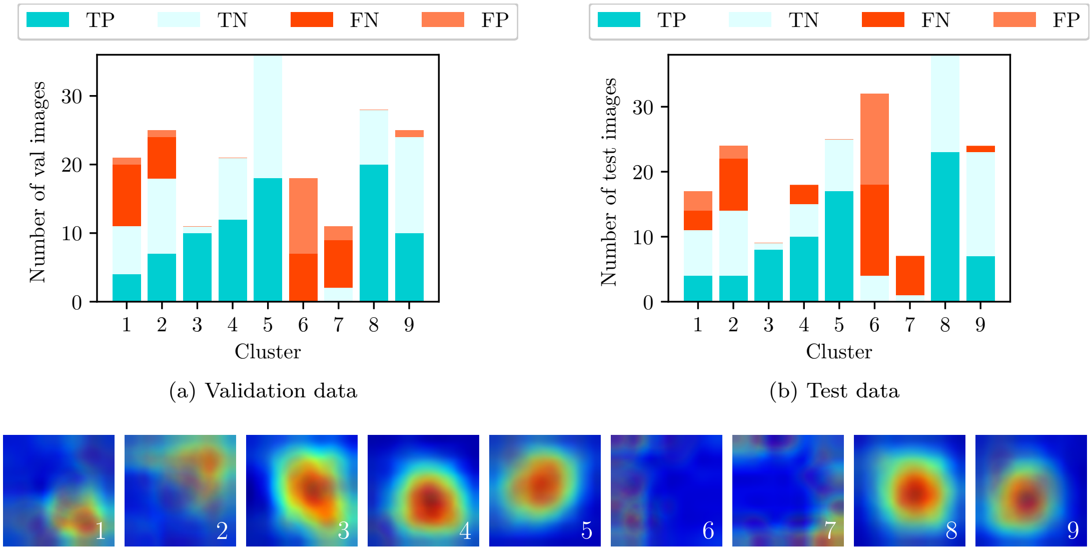
3.2.2 Local Analysis: Saliency Maps of Single Sample Inputs

In some of the resulting Grad-CAM maps, a hotspot near the center is highlighted in the image as shown in Fig. 3.3 b). In other maps, the highlighted regions are located near the edges or scattered in the image. It is easy to analyze which regions have an influence on the model's decisions since compact regions are highlighted.

A considerable amount of the OSM results resemble noise regardless of stride and patchsize for occlusion. Only a minor portion of the results show larger connected regions that are important for decision, as shown in Fig. 3.3 c). These are located in the area of the image that shows, among other things, the hidden cauliflower curd or highlighted leaf regions. Many maps show several smaller highlighted regions, which are difficult to explain because they do not indicate a unique plant trait. The ability of a simple explanation of the results varies more than forGrad-CAM.

In LIME maps, we see that the computed superpixels are not able to summarize pixels to semantically meaningful regions. This could be caused by the structure or the strong overlap of neighboring plants. Due to this, LIME saliency maps are difficult to connect to general statements about the reliability of classification outputs. An example of a sample analyzed by LIME is shown in Fig. 3.3 d). We consider LIME not suitable for our application.

Based on the assessment of single saliency maps, we consider Grad-CAM and OSM to be the most suitable approaches in our framework.



(a) Validation data

(b) Test data

(c) Prototypes

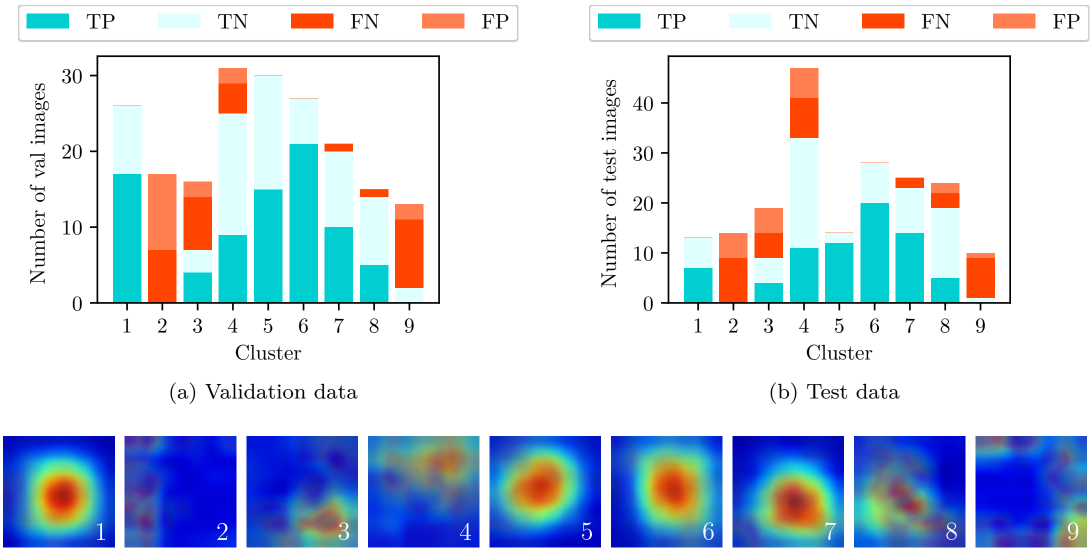
Figure 3.4: SC results of Grad-CAM maps. The absolute number of (a) validation (val) images and (b) test images per cluster. (c) shows the prototypes computed by the mean saliency map per cluster (1) – (8).

3.2.3 Global Analysis: Clustering of Saliency Maps and Reliability Derivation

We divide the global analysis into two parts, starting with the evaluation of Grad-CAM saliency maps, followed by the evaluation of OSM concerning the different types of unsupervised techniques.

Grad-CAM:

Fig. 3.4 shows the absolute number of Grad-CAM map assignments of the clustering results for 9 clusters. A distinction is made between the validation and test set. The confusion matrix entries are differentiated by color. Our experiments have shown that 9 clusters produce a good separability between false and correct predictions. Furthermore, depending on the amount of data, there are enough data points per cluster to make a reliable statement. Based on the distribution of validation data in Fig. 3.4a, it becomes evident that cluster 6 contains 100% false predictions, which are equally divided between FP and FN, while cluster 7 contains about 80% false predictions, mainly FN. This means that over 66% of all FN and FP belong to those two clusters. The remaining part of false predictions are sorted into clusters 1 and 2, where the percentage is between 20% and 40% of data samples within those clusters. The other clusters contain less than 3% false predictions. The clustering analysis allows us to say with high confidence that samples assigned to clusters 6 and 7 are equivalent to false predictions and should be



(a) Validation data

(b) Test data

(c) Prototypes

Figure 3.5: EM results of Grad-CAM maps. The absolute number of (a) validation (val) images and (b) test images per cluster. (c) shows the prototypes computed by the mean saliency map per cluster (1) – (8).

adjusted. The reliability of the classification results of the saliency maps assigned to these clusters is, therefore, low and should be disseminated to the farmer. This is underlined in particular by the cluster assignments of the test data (Fig. 3.4b). We observe that 66% of the false predicted test data are assigned to clusters 6 and 7. The proportion of false predictions in the other clusters is comparable to those within the validation data.

The prototypes of Grad-CAM maps per cluster are shown in Fig. 3.4c. More than half of the prototypes (3,4,5,8,9) highlight the region in the center of the image. This is the location in the RGB input images of cauliflower curds covered by leaves, which are the indicators of cauliflower harvest-readiness. Even though the cauliflower curd is not directly visible in the images, the model identifies the center of the plant as an essential feature for the classifier to determine the harvest-readiness. The interpretation of the classification results is straightforward and understandable for these clusters. The previously noticed clusters 6 and 7 also vary in these representations to the other clusters. In the image data assigned to these clusters, the classification model finds no distinctive features for determining the harvest-readiness. The visualization of the prototypes thus supports the model's reliability in addition to the cluster assignment since the visual representation is easier for the user to understand and interpret.

We conduct the same analysis on the Grad-CAM maps for the EM approach. The clustering results are shown in Fig. 3.5. For EM, clusters 2 and 9 are clearly distinguished by a high percentage of more than 80% false predictions per cluster, constituting about 66% of the total false predictions. The remaining false predic-

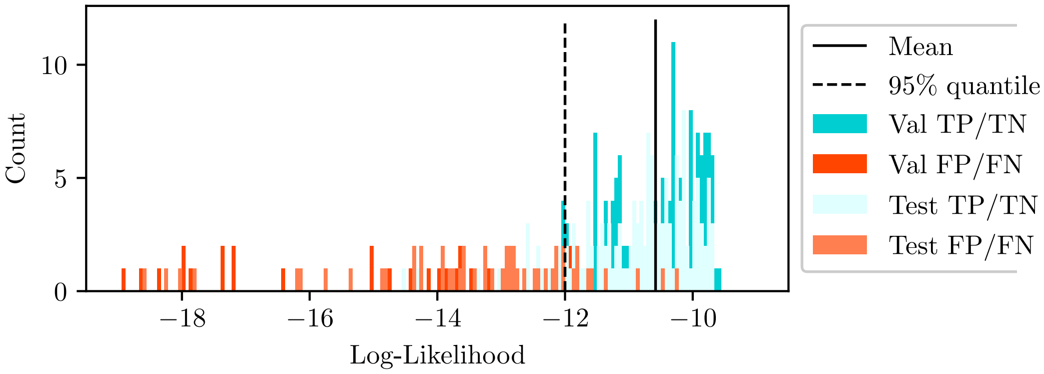


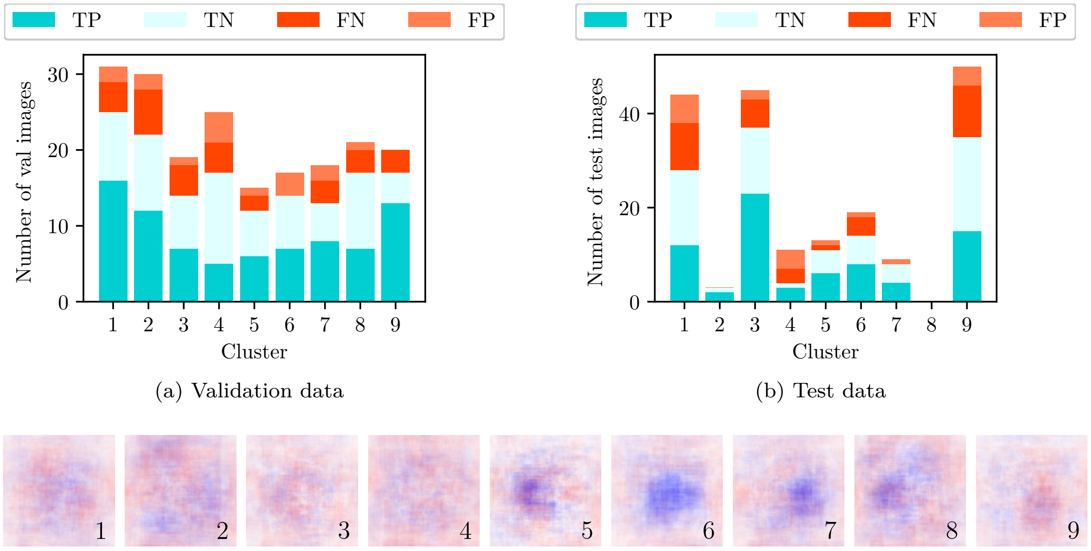
Figure 3.6: KDE results of Grad-CAM maps. Distribution of log-likelihood values for validation (val) and test data colored by assignment to confusion matrix entries. The mean and 95% quantile are based on the PDF of the true predictions of validation data.

tions are distributed among clusters 3 and 4. The other clusters contain less than 2% false predictions. Hence, the EM clustering analysis confidently identifies samples in clusters 2 and 9 as misclassifications, suggesting adjustments. Given the previous analysis, it is evident that the classification reliability of saliency maps in these clusters is low, necessitating dissemination to the farmer. again, this is underlined by the cluster assignments of the test data (Fig. 3.5b), where we observe the same proportion of false predictions in the clusters compared to those within the validation data.

The prototypes of EM maps per cluster are shown in Fig. 3.5c. We observe similar phenomena in the SC results, where clusters with a high percentage of false predictions exhibit no distinct features. Additionally, roughly half of the prototypes highlight the center, indicating the location of the cauliflower curd.

Fig. 3.6 represents the distribution of log-likelihood values computed under the PDF of the true validation data estimated using KDE. The figure illustrates the distribution of true and false predictions for both validation and test data. We observe that the distributions of true and false predictions of validation data can be largely separated within the range of log-likelihood values. This phenomenon arises due to the inability of the maps associated with false predictions to align with the distribution of true predictions, which happens due to the varying features between true and false predicted maps. 84% of the false predictions lie outside the confidence interval. Consequently, the reliability of the classification results of the saliency maps with a log-likelihood value lower than the quantile value is low and should be communicated to the farmer. Similar observations can be made for the test data. 13% of the true predictions for the test data fall within the confidence interval, while 83% of the false predictions lie outside the confidence interval.

Overall, we observe that it is feasible to assess the reliability of the predic-



(a) Validation data

(b) Test data

(c) Prototypes

Figure 3.7: SC results of OSM maps. The absolute number of (a) validation (val) images and (b) test images per cluster. (c) shows the prototypes computed by the mean saliency map per cluster (1) – (8).

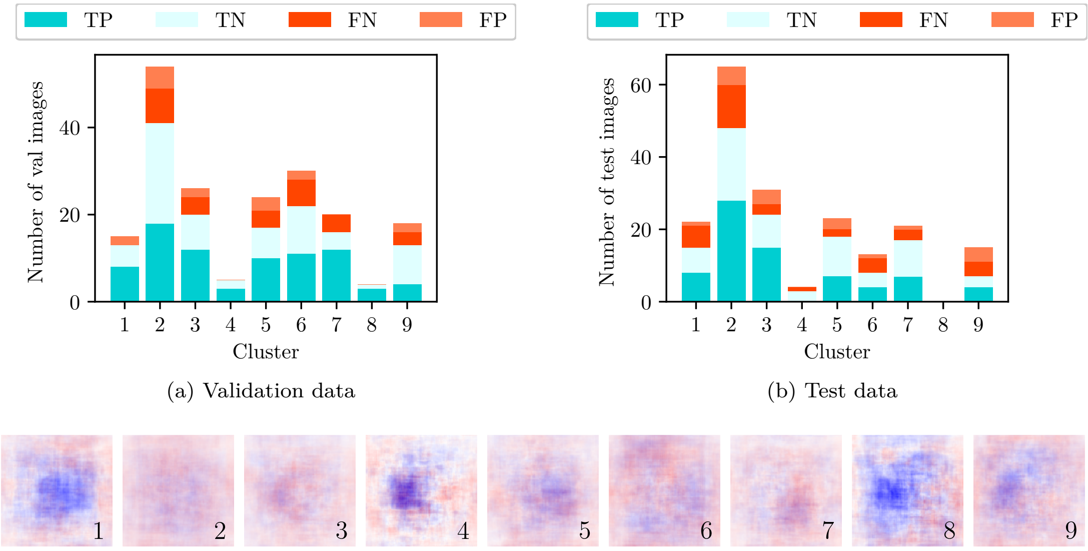
tions using Grad-CAM maps across all three methods despite differences in their underlying methodologies.

OSM:

The clustering of the OSM maps using SC shows a uniform distribution of false predictions in all clusters (Fig. 3.7a). The percentage ranges from 10% to 30%. Based on the OSM cluster results, no statement can be made about the reliability of the results. The probability that a false prediction occurs in one of the clusters is similar for all clusters. The cluster assignment of the test data shows a similar distribution (Fig. 3.7b). Only clusters 2 and 8 stand out for test data. It should be noted that the assignment to these clusters corresponds to a single image only.

The prototypes also suggest no clear trend in terms of what the model uses as an informative feature in the RGB images (Fig. 3.7c). Clusters 5 to 8 show a hotspot near the center, which, just like Grad-CAM, suggests that the model is paying partial attention to the canopy covering the curd. However, no association between true and false predictions can be established.

Similar to the clusters calculated using SC, the clustering results obtained through EM also demonstrate a uniform distribution of false predictions across the different clusters (Fig. 3.8a). Only clusters 4 and 8 exclusively contain true predictions, however, they are not representative as they only include 2 samples each. Thus, the EM results indicate that drawing conclusions about the reliability of the classification results based on the interpretability technique of OSM is not



(a) Validation data

(b) Test data

(c) Prototypes

Figure 3.8: EM results of OSM maps. The absolute number of (a) validation (val) images and (b) test images per cluster. (c) shows the prototypes computed by the mean saliency map per cluster (1) – (8).

feasible for our use case.

The prototypes for the EM clusters, shown in Fig. 3.8c, behave similarly to those from the SC approach. Clusters 1, 4, and 8 highlight the importance of central features in the OSM maps, independent of their correlation with true or false predictions.

The application of KDE on OSM maps in Fig. 3.9 demonstrates that the distinction between true and false predictions is difficult even on the basis of the distributions of the log-likelihood values. The distributions overlap within the same range of values. Specifically, 0% of the false predictions of the validation data falls outside the confidence interval. Consequently, we cannot ascertain the reliability of the classification results of the saliency maps with a log-likelihood value below the quantile value, as is possible for Grad-CAM. Similar observations can be made for the test data. In particular, 10% of the true predictions and only 2% of the false predictions of the test data lie outside the confidence interval.

Overall, we note that regardless of the unsupervised method employed, assessing the reliability of predictions with OSM maps is challenging. The key consideration lies in generating interpretable saliency maps that exhibit distinct features. Comparing the prototypes of the OSM approach with those of the Grad-CAM approach, we see that for our scenario, the Grad-CAM approach results in more interpretable maps than the ones of OSM. Since no clear differentiation between false and correct predictions can be made in the data for OSM, the adjustment step introduced in this work is only applied to the Grad-CAM results. Adjusting the classification results based on the clustering results would worsen rather than

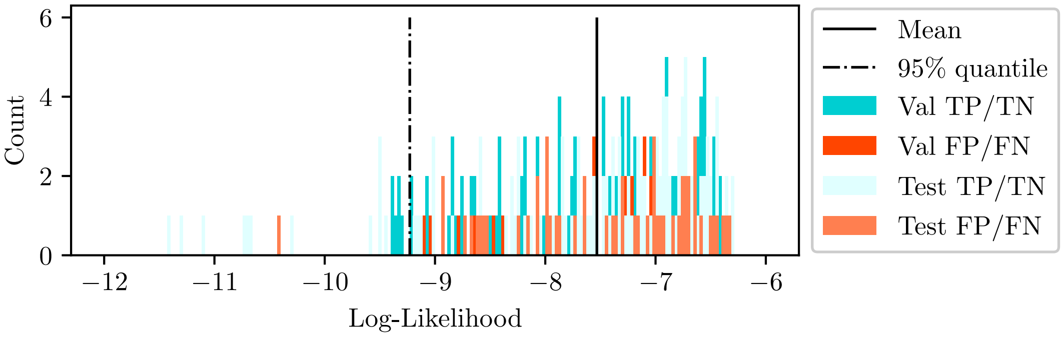


Figure 3.9: KDE results of OSM maps. Distribution of log-likelihood values for validation (val) and test data colored by assignment to confusion matrix entries. The mean and 95% quantile are based on the PDF of the true predictions of validation data.

improve the model results.

In summary, the combination of saliency map analysis and clustering provides information about the reliability of classification results. Nevertheless, careful consideration should be given to the choice of saliency mapping approach, as its influence is significant. Once a suitable saliency mapping method is identified, the visualization of prototypes not only supports the model's reliability but also aids in cluster assignment, as visual representations are more user-friendly and interpretable.

3.2.4 Adjustment of Model Predictions

Regarding the application of the adjustment step to Grad-CAM maps as explained in Fig. 3.1, we observe improvements in overall accuracy and balanced class accuracy for all three unsupervised methods. Tab. 3.1 presents the comparison between the accuracies achieved by the original model and after applying the adjustment steps. KDE achieves the highest improvement in the oaAcc and bcAcc of the validation set, with improvements of 16.84% and 16.30% respectively. The improvement achieved by SC and EM show no significant difference compared to

Table 3.1: Comparison of the overall accuracy (oaAcc) and balanced class accuracy (bcAcc) for three unsupervised methods Spectral Clustering (SC), Expectation-Maximization (EM), and Kernel Density Estimation (KDE) applied on validation (val) and test set of Grad-CAM maps.

Accuracy	ResNet-18$_3$		SC		EM		KDE	
	val	test	val	test	val	test	val	test
oaAcc [%]	77.04	72.16	89.80	87.11	90.31	83.51	93.88	86.08
bcAcc [%]	77.52	72.75	90.27	87.24	90.60	87.11	93.91	85.84

each other, with improvement ranging from 12.76% to 13.27% for oaAcc and from 12.75% to 13.08% for bcAcc. For the test data, the SC approach yields the highest accuracies, showing an improvement of 14.95% in oaAcc and 14.49% in bcAcc. On average, the results of KDE are subsequent, followed by those of EM.

Nevertheless, it is worth questioning which of the unsupervised approaches, in combination with the adjustment step, is the most suitable. Although the application of KDE leads to the highest accuracies in the validation set, the accuracies may also deteriorate depending on how well the correct and false predictions can be distinguished in the feature space, as illustrated by the negative example in Fig. 3.9. We assume that at least 5% of the true predictions are adjusted without knowing the ratio of false to correct predictions below the quantile. For SC and EM, due to the approach of calculating the reliability scores, it is ensured that always more false than true predictions are adjusted, thus the accuracy never deteriorates compared to the original model's accuracy. For dissemination to the farmer, we recommend using one of the clustering-based approaches, as these are the safer options. Furthermore, we are able to generate a more detailed reliability score for the clustering methods in range $[0, 1]$, while for KDE, we provide only a score $r \in \{0, 1\}$.

3.3 Conclusion

This work proposes a framework to derive a reliability score for cauliflower harvest-readiness estimations that operates post-hoc during inference time without the need for human interaction. Our work combines a ResNet-18 classification model with an unsupervised approach for group assignments of saliency maps using Spectral Clustering (SC), Expectation-Maximization (EM), or Kernel Density Estimation (KDE) to derive a reliability statement of classification predictions. Since the reliability value is in a fixed range between 0 and 1, it is intuitive and can be provided to the farmer as a decision support. In addition, the classification predictions can be adjusted, and the accuracy can be improved. We compare three saliency mapping approaches: Gradient-weighted Class Activation Mapping (Grad-CAM), Occlusion Sensitivity Mapping, and Local Interpretable Model-agnostic Explanations, and the three above-mentioned unsupervised approaches: SC, EM, and KDE. The combination of Grad-CAM and SC proves to be the most useful in our scenario.

For our use case, our approach enables the correct harvest-readiness estimation on GrowliFlowerR, a subset of the GrowliFlower dataset, of approximately 9 out of 10 cauliflowers, compared to the state-of-the-art approach ResNet-18, which achieves only approximately 3 out of 4 correct predictions. Our framework offers the advantage of not requiring any interaction with the training process. It can be applied to already trained models without accessing or modifying the model architecture. We provide interpretable visualizations and a reliability score for the model's decision. Since our framework only considers false predictions, the approach can also be used to disseminate reliability in multi-class tasks.

Chapter 4

Multi-class Harvest-readiness Classification

The binary classification approach presented above only indicates whether a plant is ready for harvest or not, lacking the detailed temporal information needed for optimal operational planning. By classifying the exact harvest day, we gain finer results in predicting the harvest time directly. This enables more precise planning and resource allocation for harvesting. We extend the idea of binary classification of harvest-readiness to a multi-class classification problem by attempting to directly determine the specific harvest day, following scenario 3. For implementation, we again examine the two classification models, ResNet-18 and ViT-B/16 and train models for different forecasting times.

Three primary expectations drive our experiment. Firstly, we anticipate lower accuracies compared to binary classification due to the increased complexity of the task and the smaller amount of data available per class. This increased complexity arises from the need to distinguish between multiple classes (specific harvest days) rather than just two states (ready or not ready for harvest). Consequently, the data for each individual class (specific harvest day) is less comprehensive than the data used in binary classification, likely leading to lower overall classification accuracy. Secondly, we hypothesize that the application of the ResNet-18 model will yield higher balanced class accuracies compared to ViT-B/16. This is due to ResNet's ability to handle the significantly reduced dataset more effectively, which contrasts with the binary classification approach where both models achieved similar accuracies. Furthermore, we expect that models, unlike in the binary case, will exhibit higher accuracies for shorter lead times. These differences are likely because the models are constructed using data from only one specific day of acquisition, thereby reducing their dependency on exposure and other varying temporal conditions. We aim to determine which developmental states are more suitable for predicting the harvest day in advance.

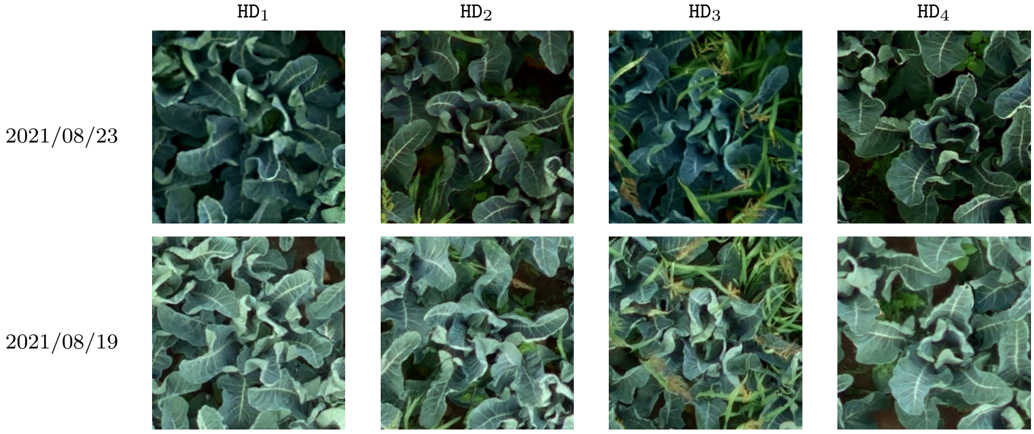


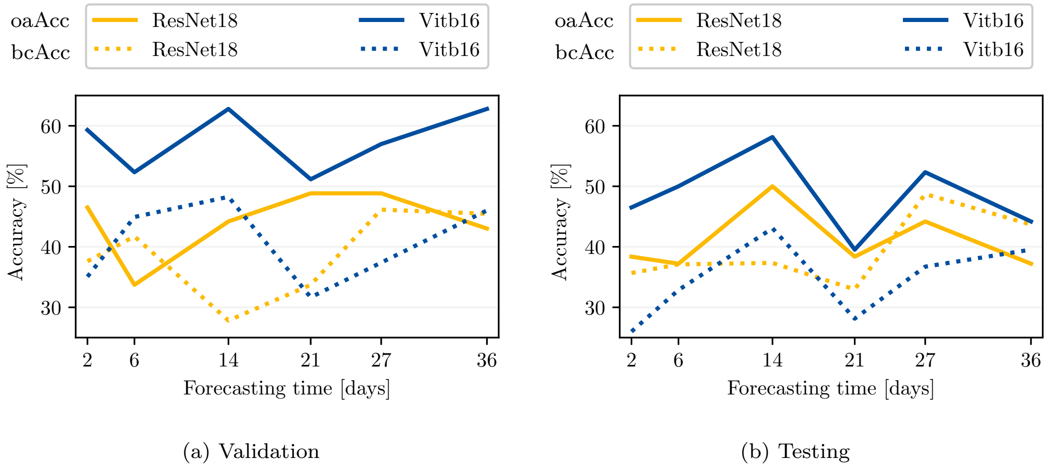
Figure 4.1: The images show plant examples acquired at two days and labeled with the harvest day (HD) indicated in the columns: HD$_1$, HD$_2$, HD$_3$, and HD$_4$.

4.1 Experimental Setup

For this experiment, we use images from the GrowliFlowerR dataset acquired on field 2. We utilize multiple subsets, each comprising all images from a specific acquisition day with a lead time of $t \in \{2, 6, 14, 21, 27, 36\}$ days before harvest, where t corresponds to the acquisition dates {2021-08-23, 2021-08-19, 2021-08-11, 2021-08-04, 2021-07-29, 2021-07-20}. The data in each subset is grouped into four classes: HD$_1$, HD$_2$, HD$_3$, and HD$_4$.

A total of 500 reference plants are available, of which only 470 plants were harvested over four days. The remaining 30 plants were referenced as not ready for harvest and are therefore excluded from the dataset for this experiment. We apply the following procedure to each of the subsets. The division into training, validation, and test data is performed as described in Sec. 3.2.3. This results in a training set of 298 plants and validation and test sets of 86 plants each for both subsets. To extend the datasets, we perform standard augmentations, ensuring that each class is equally represented post-augmentation to balance class distribution. After augmentation, each subset comprises 2892 training samples. Fig. 4.1 represents images for each of the four classes across two acquisition days, highlighting the difficulties in accurately classifying the harvest day. This challenge arises from the similar developmental stages of the plants, making differentiation more complex.

For our classification models, we select ResNet-18 and ViT-B/16 for single input images. Both models use a final linear layer of length four to address the four-class problem. For each architecture, we train one model with a forecasting time of $t \in \{2, 6, 14, 21, 27, 36\}$. The training for each model consists of at least 50 epochs and stops if the validation accuracy does not significantly increase over 10



(a) Validation

(b) Testing

Figure 4.2: Comparison of ResNet-18's and ViT-B/16's model accuracies computed for different forecasting times of the harvest day. The overall accuracy (oaAcc) and balanced class accuracy (bcAcc) for both (a) validation and (a) test sets are illustrated.

epochs. We use an Adam optimizer and fine-tune the weight decay in a range of 0.01 and 0.0001. We also fine-tune the starting learning rate in the range of 0.001 and 0.0001 and reduce it while training using a learning rate scheduler with a step size of 5 and a factor of 0.1. We utilize oaAcc and bcAcc as evaluation metrics.

4.2 Experimental Evaluation

As expected, we achieve lower accuracies for the multi-class problem for both the ResNet-18 and ViT-B/16 models compared to the binary case, due to a smaller dataset and increased task complexity. In Fig. 4.2, we show the comparison between ResNet-18's and ViT-B/16's model accuracies computed for different forecasting times of the harvest days. ViT-B/16 achieves up to 63% in oaAcc, which is higher than ResNet-18's maximum of 50% on both the validation and test sets. Since bcAcc is an accuracy metric that averages the accuracy per class, it is particularly important for multi-class classification problems, especially when the data samples per class are unevenly distributed. Therefore, we attribute greater significance to the bcAcc. The bcAcc is similar for both models. Only for forecasting times of 14 and 27 days do the values vary by 8%, indicating that ViT-B/16 may achieve better results for forecasting times up to 21 days, while ResNet-18 performs better for forecasting times beyond 21 days.

Contrary to our expectations, both types of models do not exhibit higher accuracies for shorter lead times. One reason for the difficulty in predicting the harvest day could be the occlusion caused by neighboring plants or weeds. Accurate predictions become more challenging as the plants develop further because the leaves of neighboring plants overlap more, making it harder to identify specific features in the images. Additionally, some areas of the field are infested with weeds, which are particularly present in the later acquisition days and thus at shorter lead times.

We cannot definitively attribute the fluctuations in accuracies, as this would require deeper analyses that we do not discuss further in this work but would be interesting to explore in future research. Nevertheless, we have two hypotheses regarding the causes of these fluctuations. The first hypothesis relates to the image quality, which, unlike in the binary case, does not bias the decision towards a particular class but still struggles to differentiate between classes, e.g., due to an increased proportion of blurry or dark images. Another possible reason is that, at certain times, the plants may exhibit differences in development, which simplifies classification and is reflected in the classification accuracy.

The analysis of this scenario indicates that it is possible to estimate the harvest day from a given set of days based on the given data. Assuming that fields exhibit similar developmental patterns, we assume a transferability to other fields. However, due to the strong weather dependency of growth, we consider the application to new data with the aim to achieve comparable accuracy to be challenging for the time being.

4.3 Conclusion

We classify the harvest day of cauliflower plants using a CNN-based ResNet-18 classification model and an attention-based Vision Transformer model ViT-B/16, considering different forecasting times for harvest-readiness. The comparison between the accuracies achieved by the two models shows that the ViT-B/16 model achieves higher overall accuracies and balanced class accuracies for forecasting times up to 21 days. In contrast, ResNet-18 demonstrates better-balanced class accuracies for higher forecasting times.

The transformer-based network shows greater potential for determining the harvest day, however, the data foundation must be expanded. More data needs to be labeled, and methodologies such as self-supervised learning should be utilized to handle datasets with a small proportion of labeled data efficiently. In our case, self-supervised learning approaches can leverage unlabeled data from the GrowliFlowerT dataset to pre-train model weights on unlabeled cauliflower image data.

Part III

Image Time Series Analysis

Introduction

Determining harvest information of cauliflower from UAV images of single points in time has already been done in Part II. Opposed to individual time points, which can only capture a plant's current state to a limited extent, time series, enabling continuous monitoring of the entire growth cycle of plants, offer insights into the dynamic and current rate of plant development. This facilitates the comprehensive analysis of growth patterns and the estimation of crop yields. Utilizing image time series has already shown great success in the field of satellite data, e.g., for crop type mapping [95], [219], [220] or yield prediction [221]–[223]. Therefore, using time series shows a high potential for improving the accuracy of harvest prediction using UAV images.

UAV data acquisition and processing on a weekly or even daily basis is time-consuming. Optimization through the reduction of low-quality data enables model improvement, as this data harms the result [224]. Low-quality data results, e.g., from time points that are less relevant or have no information gain for model predictions [225]. Thus, finding time points that contribute most to a correct harvest-readiness estimation is crucial to improving the model and resources like time and money for future observations.

For our study, we classify cauliflower plants according to their harvest-readiness using image time series showing plants' development over time. A modified ResNet18 [139] classifier is employed. We compare models using images of single time points shortly before harvest [8] to those using time series with initial acquired time points without explicit selection. Furthermore, we use the explainable ML method GroupSHAP [209] to identify which image time points contribute most to the model's prediction, allowing us to selectively determine time points that increase the model's accuracy. We compare the time points with the respective development stages of the plants. From this, we conclude which developmental stages are generally important to determine harvest-readiness and propose how to reduce data acquisition resources.

As the main contributions of this part, we show that:

- the use of time series compared to single time points lead to an improvement in the predictive accuracy of cauliflower harvest-readiness up to 4%;
- the use of GroupSHAP helps to select time points to improve the accuracy by a further 4% up to 89%. This information can be connected to growth stages and used to reduce the required resources for data acquisition in future works.

This part is based on our published paper by Kierdorf et al. [9].

Chapter 1

Scenarios

All previous scenarios consider single time points as the basis of data, corresponding to individual growth stages of the plants. These approaches are applied in Part II and evaluated with respect to our objectives. The advantage of field monitoring is that we obtain data throughout the entire growth period and can integrate it into our prediction of harvest-readiness. Part III addresses the question of whether time series information leads to an improvement in prediction accuracy. For the investigation of this objective, we consider three additional scenarios. For clarity, we continue numbering the scenarios sequentially.

Scenario 4

Scenario 4 is an extension of scenario 2. It addresses whether a plant will be ready for harvest in t days from day T_h, but based on the temporal information available during growth. To aid comprehension, the yellow box visualizes this in Fig. 1.1, which encompasses multiple time points. Like scenario 2, this scenario involves a binary problem. In scenario 4, it can be investigated which length of a time series provides the most accurate accuracies.

Scenario 5

Scenario 5 extends scenario 3 by incorporating time series information prior to the first harvest, as depicted in Fig. 1.2. Data acquisition during the harvest is omitted, and the investigation focuses on determining the harvest time point T_h for a plant. Similar to scenario 3, this scenario is approached as a multi-class classification problem.

For both scenarios utilizing time series information, we observe that the workload of data acquisition, processing, and analysis increases compared to scenarios that utilize single time points. However, it is desirable that the prediction results

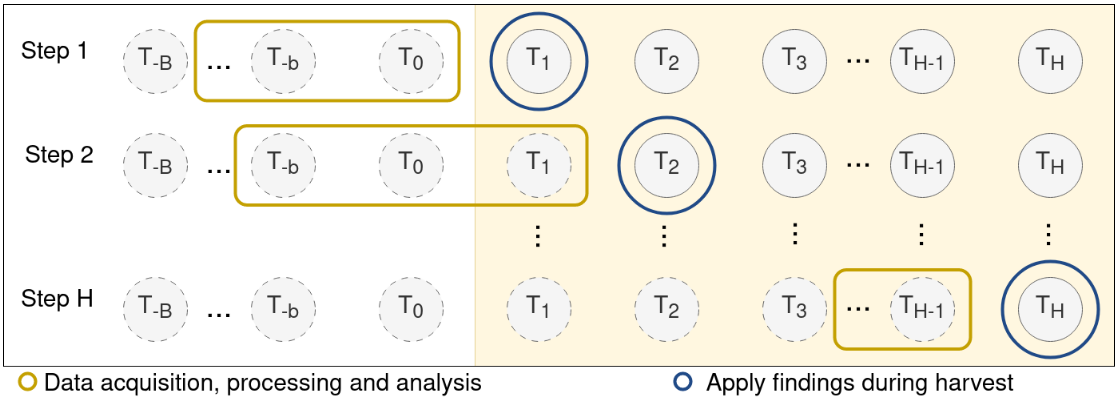

Step 1
Step 2
Step H
T_{-B}
T_{-b}
T_0
T_1
T_2
T_3
T_{H-1}
T_H
Data acquisition, processing and analysis
Apply findings during harvest

Figure 1.1: Scenario 4 visualizes the interaction between data collection, processing and analysis with the application of tactical management decisions for the binary classification of harvest-readiness based on time series data. In scenario 4, the actions are conducted at multiple time points up to a lead time of t days before each harvest.

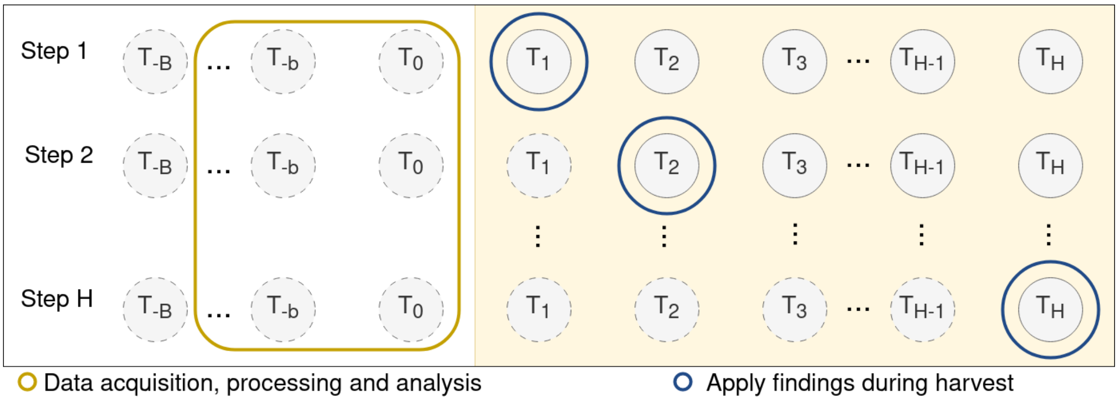

Step 1
Step 2
Step H
T_{-B}
T_{-b}
T_0
T_1
T_2
T_3
T_{H-1}
T_H
Data acquisition, processing and analysis
Apply findings during harvest

Figure 1.2: Scenario 5 visualizes the interaction between data collection, processing, and analysis by applying tactical management decisions for the multi-class classification of harvest-readiness based on time series data. In scenario 5, data collection, processing, and evaluation are conducted at multiple time points up to t days before the first harvest date T_1.

improve with the additional information about plant development. To reduce the workload, it is worthwhile to explore whether there are specific time points or developmental stages that positively influence classification accuracy.

Chapter 2

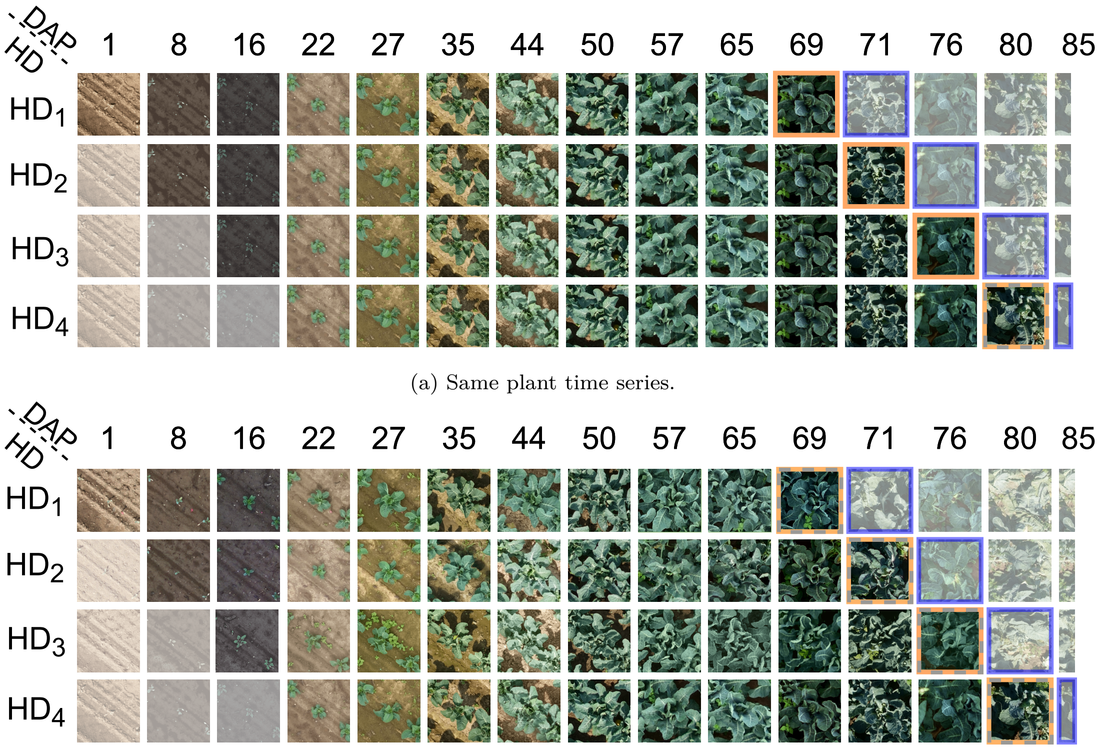
Classification based on Initial Time Series

For classification based on single time points, we assume that the time interval between image acquisition and harvest must be kept short, as factors such as weather still change the development considerably [226]. No prior knowledge about previous plant development is given in this case. In this experiment, we want to investigate whether the use of time series information for the classification of harvest-readiness as described in scenario 4 is more beneficial than the use of individual time points, investigated in Part II because the use of time series integrates the temporal development of the plants into the model. We also address whether it is worth integrating early acquisition times to increase model accuracy or whether it is sufficient to use time points close to harvest.

2.1 Experimental Setup

We use image time series data from field 2 of the GrowliFlowerR dataset introduced in Sec. 3.2.3, showing the development of cauliflower from planting to harvest. The dataset contains information about planting and harvest day for each cauliflower plant. The planting day is used to derive the day after planting (DAP) for each image in the time series, which represents the age of the plant. Harvesting took place on four dates. The images in the dataset are georeferenced and have the same resolution and scale. Due to different weather conditions at different DAPs, factors such as exposure and soil irrigation differ at various points in time.

We prepare the data in the same way as for our experiments based on single input images, described in Chap. 2. For this task, we used images right before harvest, as shown in Fig. 2.1 highlighted in orange, and divided into the classes `Ready` and `Not-ready` for harvest. For this experiment, we refer to these images as basic images. For our time series classification approach, we extend the basic



(a) Same plant time series.

(b) Different plant time series.

Figure 2.1: Visualization of cauliflower image time series, with a length of $T = 11$, presenting various potential harvest days HD, indicated by the blue frames, whereby a row illustrates an individual time series. (a) shows an example of generating multiple time series for an individual plant. In this example, a plant is observed and labeled as `Not-ready` for harvest on days HD$_1$, HD$_2$, and HD$_3$, shifting to `Ready` for harvest on harvest day HD$_4$, indicated by the grey dashed frame. Each time series is shifted by one image for HD$_1$ to HD$_4$, reflecting the progression over time. The variability in potential harvest days results in differences among the basic images within the time series, indicated by the orange frame. Consequently, the corresponding images at day after planting (DAP) also shift accordingly. Only non-transparent images are utilized as input for constructing the time series. For plants deemed `Ready` for harvest on HD$_1$, there exists only one plausible time series since harvesting occurs on that specific day. Equivalently, for plants harvested on HD$_2$ and HD$_3$, there are two and three conceivable time series, respectively. This method of generating time series remains applicable across varying time series lengths. (b) shows time series of four different plants representing the four harvest days. This illustration compares `Ready` and `Not-ready` for harvest plants. Figure source: Kierdorf et al. [9].

images by $T - 1$ images acquired chronologically before the used basic image, resulting in a time series with T individual time points. Each image within the time series represents a different developmental stage of the plant. We vary T for later experiments with $T \in \{1, 2, \ldots, 11\}$, resulting in time series with different temporal lengths. We denote these time series as initial time series (iTS).

Example image time series with a length of $T = 11$ for one specific plant are shown in Fig. 2.1a. The presented plant is observed and labeled as `Not-ready` for harvest on harvest days HD$_1$, HD$_2$, and HD$_3$, shifting to `Ready` for harvest on harvest day HD_4. Each time series corresponds to one of the harvest days. If a plant is classified as `Not-ready` for harvest on a given harvest day, it is reclassified for the next harvest day. The variability in potential harvest days results in differences among the baseline images within the time series, indicated by the orange frame. As we align the time series with these baseline images, the temporal start and end points of the series shift towards the harvest day, depicted by non-transparent images. For plants deemed `Ready` for harvest on HD_1, there exists only one plausible time series since harvesting occurs on that specific day. Equivalently, for plants harvested on HD_2 and HD_3, there are two and three conceivable time series, respectively. This method of generating time series remains applicable across varying time series lengths. Thus, we can generate up to four time series for a specific plant, dependent on the harvest day. The images are aligned by DAP and illustrate which DAP is used for classification regarding the potential harvest days. Since the basic images were taken on different DAPs depending on the potential harvest day, iTS contain different stages of development. Fig. 2.1b compares four different plants, each labeled with a different harvest day. We use information about available developmental stages of cauliflower according to Feller et al. [211], as listed in Tab. 2.1. For this experiment, we relate the developmental stage to the DAP.

We compare three types of models to investigate whether using time series information to classify harvest-readiness is more beneficial than using individual time points. In the first model, we use the ResNet-designed model structure for time series data but use single time points as input and denote this model as our baseline. As a reference to the baseline, we use the ResNet18_3 model for single image inputs without the additional linear layer, investigated in Part II. For both models, we use the basic images as input. As the third model type, we use our designed network and iTS as input. For all model types, we calculate the oaAcc and bcAcc and compare them across the different types of models.

We train one time series model for each input time series length T. We normalize the input images before feeding them into the model. The training for each model consists of at least 60 epochs and stops if validation accuracy does not increase significantly over 10 epochs. We use a batch size of 16 and the Adam optimizer with a learning rate of $1\text{e}{-5}$. The learning rate is reduced using a scheduler with a step size of 20 and a factor γ of 0.1. We adjust the weight decay and linear layer mentioned earlier for each model through hyperparameter tuning. We consider weight decays α in the range of $[1\text{e}{-1}, 1\text{e}{-3}]$ and scaling factors λ in the range of $[2, 4]$. As the final model of training, we select the model with the highest validation accuracy. For reproducibility, we set all used seeds to 0. We run our experiments on an AMD EPYC 7742 64-Core processor and an NVIDIA A100 PCIe graphic card with 40 GB hBM2 RAM. The runtime of the model with the

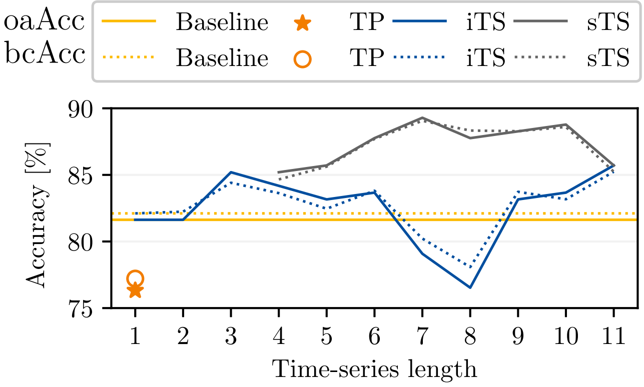


Figure 2.2: The plot opposes the baseline model accuracies for single time point inputs to the accuracies achieved in our further investigations in Part II (TP), to our computed initial time series (iTS) accuracies, and to the selective time series (sTS) accuracies. The plot shows the overall (oaAcc) and balanced class accuracy (bcAcc) for different time series lengths. For sTS, time points are excluded starting from right to left. Figure source: Kierdorf et al. [9].

most input features with $T = 11$ is 14 minutes.

2.2 Experimental Results

The comparison between the accuracies of our baseline and the reference for single time point classification investigated in Chap. 2 of Part II shows that incorporating an additional linear layer into the model leads to a general improvement in the achieved accuracies for single time point inputs (see yellow lines in Fig. 2.2 compared to orange markers). Furthermore, we find that using iTS input data enhances model accuracy compared to the baseline in nearly all cases. When adding successive time points, we achieve higher accuracies for seven out of ten time series models, a similar accuracy in one case, and lower accuracies in two cases compared to our baseline. The increasing trend is noticeable initially but decreases between $T = 4$ and $T = 8$ and then increases again at $T = 9$ to reach a maximum value of 85.7%. The maximum increase in accuracy compared to the baseline is approximately 4% for a time series length of $T = 11$.

2.3 Discussion

We demonstrate that the use of time series information enhances the predictive accuracy of the model, even when the cauliflower curd is not visible in any image within the series. Kierdorf et al. [8] demonstrate that it is possible to determine harvest-readiness even when the curd is occluded by the canopy. Using explainable machine learning through the Grad-CAM technique, they showed that the

ResNet18 model's decision is influenced by specific image features. These features primarily include the leaves at the center of the plant, which protect the curd. Since we also use a ResNet18-based model architecture, we expect these insights to be applicable to time series data. Furthermore, information on the plant's development over time provides additional features that are utilized for predicting harvest-readiness and thus increase the accuracy.

We attribute the decrease in accuracy to the fact that not every time point in the data set provides relevant information to the model. Some time points may exhibit redundancy or correlation and share the contribution to the output. Generally, this could be because there is no significant visual growth of the plants between two acquisition days. Particularly in the later stages of development, the plants no longer grow visibly but continue to develop the curd internally. Another reason could be that additional time points negatively impact the accuracy by confusing the model. This may be due to irrelevant features or noise in the data [225], such as slightly blurry images, that occur when processing the raw data into orthophotos [7]. By examining the time points added for specific time series lengths, we find that time points within the DAP interval $[44, 65]$ are more likely to harm the accuracy. The increase in accuracy can be attributed to adding new informative features by adding additional images.

2.4 Conclusion

In this experiment, we classify image time series of cauliflower plants, depicting the temporal development concerning their harvest-readiness. For this purpose, we use a ResNet18 model as an encoder and integrate the plant age through positional encoding to improve the discrimination between young and underdeveloped plants. In our experimental investigations, we demonstrate that models based on image time series data exhibit superior accuracy than the baseline model, which only considers a single time point as input. This improvement can be attributed to the integration of additional plant development information.

Chapter 3

Classification based on the Selection of Time Points with GroupSHAP

In this experiment, we investigate how single time points within a time series contribute to the classification result and how excluding time points affects the model accuracy. The basis for this experiment is built on scenario 4. Literature has shown that excluding features (here time points) based on feature selection can improve the model accuracy [227]–[229]. We connect the time points with the BBCH developmental stages according to Feller et al. [211] and investigate with GroupSHAP whether certain developmental stages have a low contribution to model accuracy and can, therefore, be omitted from data collection to conserve resources.

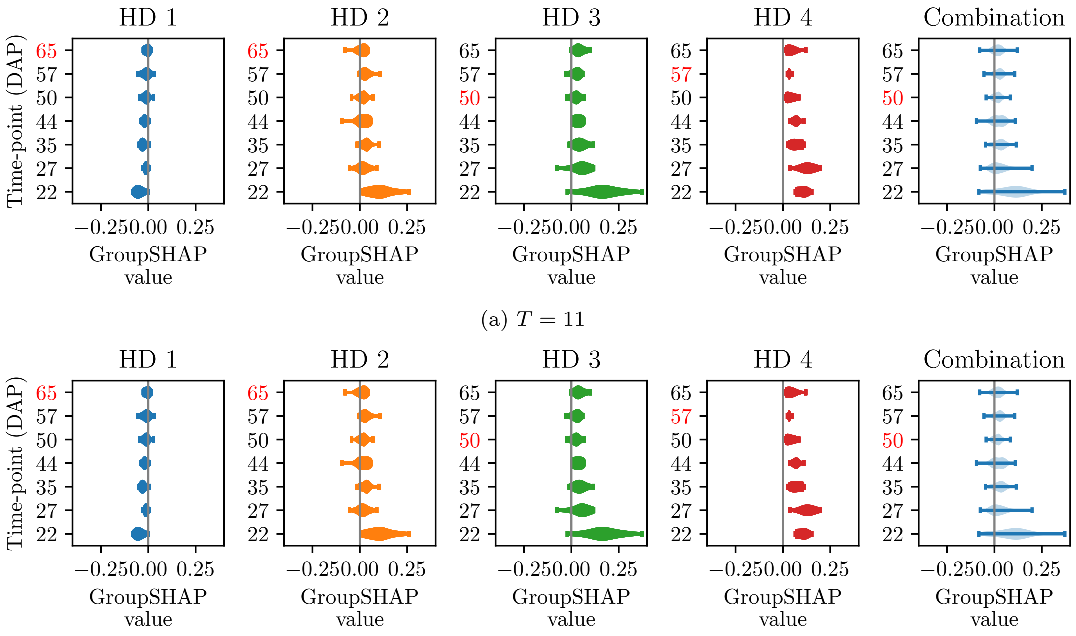
3.1 Experimental Setup

This experiment is a follow-up experiment on the time-series investigation. We use the same data, model architecture, and training setup as used in the previous experiment described in Sec. 2.1. We take the iTS model with $T = 11$ calculated in our first time series experiment and (i) calculate the entity contribution of the time points using GroupSHAP. We (ii) exclude the time point with the lowest mean absolute GroupSHAP from iTS overall harvest days since it has the most neutral contribution (closest to 0). In theory, the day with the lowest contribution would have to be excluded separately for each HD to receive the highest model accuracy, as different DAPs are contained in the time series of the different HDs. In practice, however, concerning resource-saving data acquisition, not only selected parts of the field are flown over, but the entire field, so that certain points in time must be completely excluded. Therefore, we exclude the time points with the lowest mean absolute contribution across all time points and, thus, exclude the mean macro developmental state over the whole field. We denote the new time series with the selected time points as sTS. Next, we calculate (iii) a new model using the sTS and recalculate the accuracies. We repeat (i)-(iii) using the most recently determined sTS instead of iTS.

We specify that the first three and last four acquisition days are always included in the time series. Keeping the last four acquisition days is important because it allows us to determine whether the class `Ready` or `Not-ready` for harvest can be derived in the coming days. Without these time points, there is no reference point for predicting harvest-readiness. If we classify a plant as `Ready`, it will be ready for harvest within the coming days, i.e., the last image in the time series is the last one before harvesting. Including the time points close to harvest has proven to be beneficial in maintaining stable results despite weather fluctuations. Another reason for always including these seven time points is to minimize data bias towards a specific HD and maintain similar data for all models, we only consider time points for exclusion where an image can be excluded from each HD. The excluded days show, on average, the same developmental stage per time point (see Tab. 2.2). Different plant developments average out over the entire field. We assume that this will also be the case for the following growing seasons. For the experiment, fixing the seven time points allows only the calculation of sTS for time series length $T \in [4, 10]$.

3.2 Experimental Results

Comparing the sTS versus iTS model accuracies, we achieve higher accuracies at all time series lengths (Fig. 2.2). Compared to the best iTS model with $T = 11$, the accuracies maintain a similar or higher level with the exclusion of selective entities. The oaAcc and bcAcc for sTS models have their maximum at 89.3% and 89.1% for a time series length of 7. This indicates that excluding specific DAPs leads to a positively developing accuracy. Thus, we observe that GroupSHAP helps to select relevant entities to increase model accuracy. The sTS model accuracies perform with lower accuracy for shorter time series ($<$ 6TPs), indicating that informative entities with a positive overall impact on the accuracy are excluded. This is because our approach uses the criterion of the lowest mean absolute value for exclusion. However, the lowest mean absolute value can also make a positive contribution to the predictions, which means that the exclusion results in a reduction in accuracy. The exclusion of entities, therefore, only makes sense up to a certain point to achieve the best model accuracy. Furthermore, we observe that a sTS of length 5 achieves the same accuracy as using 11 initial time points. This indicates that feature selection can save time and costs in data acquisition and processing to



(a) $T = 11$

(b) $T = 10$

Figure 3.1: Visual example of GroupSHAP values for time series lengths T of (a) $T = 11$ and (b) $T = 10$. The fixed time points are not shown, as they are not excluded. One violin plot shows the distribution of GroupSHAP values per time point, more explicitly per day after planting (DAP). The first four plots represent the set of GroupSHAP values classifying data of harvest day (HD) 1 to 4. The light blue plot represents the combination of the four sets. The red-marked DAPs represent the days with the lowest mean absolute GroupSHAP value. The red-marked number in the combination plot is excluded in the next selective time series model. Figure source: Kierdorf et al. [9].

obtain the same result as acquiring data over the whole growing period.

Fig. 3.1 shows the distribution of GroupSHAP values for sTS lengths with $T \in [10, 11]$. For a deeper analysis, the GroupSHAP values are separated for the potential harvest days. In addition, a total overview of a combination of all HDs is given in light blue. All five plots are related to class `Ready` for harvest. Independent of the time series length, the model tends to classify `Ready` for harvest for later potential harvest days, as the number of plants ready for harvest increases over time. This also applies to shorter time series lengths. Since, in practice, the entire field is flown over, and it is only worth eliminating an entire acquisition day, we only consider the combination plots concerning excluding time points. It turns out that on average, for $T = 11$, DAP 50 (macro stage 40) and for $T = 10$ DAP 65 (mean macro stage 43) must be excluded. The order in which the DAPs are excluded continues with 57, 44, 22, 27, and 35. From a biological perspective, the initial stages of sorting occur when the plant is in the phenological development of macro stage 4 and micro stages 1-3 (according to BBCH by Feller et al. [211]). During these stages, curd development occurs, and the curd starts growing, reaching a diameter of up to 6 cm. In the corresponding image data, there are minimal visual changes compared to earlier images, as the growth happens internally within the plant. The current appearance of the plant, which is used in the model's decision-making, is therefore determined from the images that display the most robust plant development. In contrast, the development stages of the days with the highest contribution (DAP 22, 27, and 35) are at the beginning of macro stage 3, when the main shoot begins to develop. Taking a closer look at the first two excluded acquisition time points, 50 and 65, we observe that these time points occur more frequently in the database for sTS models of length 7 and 8. Since these time points do not have a supposed positive contribution to the predictions, this explains the drop in the iTS curve.

From our observations, we conclude that the selective choice of time points improves the model's accuracy and can reduce the effort in data acquisition in the future. We obtain the best model using sTS with seven time points within a time series with oaAcc of 89.3% and bcAcc of 89.1%. We achieve an oaAcc of 76.3% and bcAcc of 76.7% with the same model on a test set. Weighing the effort of data acquisition against achievable model accuracy, we achieve an oaAcc of 85.2% and bcAcc of 84.7% on validation data and oaAcc of 78.9% and bcAcc of 78.9% on the test set when using 4 time points.

3.3 Discussion

Based on the accuracy curves for iTS and sTS, we observe that the exclusion of specific DAPs improves accuracy. This demonstrates that GroupSHAP effectively selects relevant entities to enhance model accuracy. However, excluding too many features can result in the loss of valuable information essential for accurate predictions; thus, it is important to limit exclusions to maintain higher accuracies. In addition, we have identified that shorter time series, including selected time points, yield similar accuracies compared to longer time series without time point selection. This suggests that feature selection can reduce time and costs in data acquisition and processing while achieving the same results as acquiring data over the entire growing period.

We note that selected time points are those where the model's decision-making relies on images that exhibit the most distinct visual plant development and where the main shoot begins to develop. In contrast, time points characterized by continued internal head development within the plant and less external growth are excluded. The frequent presence of acquisition time points 50 and 65 in the database for sTS models of lengths 7 and 8, which do not contribute positively to predictions, may explain the decline in the iTS curve as well. For iTS, we hypothesize that certain time points negatively impact accuracy due to irrelevant features or noise in the data, such as slightly blurry images. This insight also applies to sTS models and might be another explanation for the low contribution of the time points.

It is important to note that our statement regarding the order of DAP exclusion may change depending on the development of the plants in response to external conditions. If the field develops on average one week earlier, this shift applies to the entire field, resulting in a corresponding adjustment of all harvest days and development stages. When generalizing to other fields and farms, it is important to consider the development stages rather than solely relying on the DAP time point. Although we have not yet tested the trained model on another cauliflower farm, preliminary results indicated that the available data in the field of cauliflower harvest-readiness estimation is not sufficient to generalize and transfer the classification model to other fields. The effects of varying weather, lighting, and irrigation must be accounted for to ensure generalizability. However, altering colors in the HSV color space to simulate changes in exposure and soil conditions can inadvertently modify the perceived biological properties of the plants. For instance, a color change might make healthy leaves appear diseased, or conversely, diseased leaves appear healthy.

GroupSHAP provides valuable insights, yet there are limitations that need to be addressed. One of the primary constraints is the high complexity and substantial computational time. The method evaluates the contribution of each feature across numerous permutations, resulting in high computational complexity, particularly with large datasets and complex models. To mitigate this issue, parallel computation on multiple GPUs can be utilized. By distributing the computational workload across multiple GPUs, the time required for processing can be significantly reduced. Additionally, the method is sensitive to data quality. The explanations generated by GroupSHAP heavily depend on the quality of the input data. Noisy, incomplete, or biased data can lead to incorrect attributions and interpretations. However, GroupSHAP can also be utilized to identify such issues within the data, as this data should make a small contribution to the final prediction. High data quality from the beginning can be ensured through in-depth data cleaning and validation to guarantee accurate and reliable results.

3.4 Conclusion

We use the interpretation technique GroupSHAP to investigate the contribution of single time points within a time series to the model's prediction of cauliflower harvest-readiness and how excluding time points with the lowest mean average contribution affects the model's accuracy. We show that the explainable machine learning method GroupSHAP effectively facilitates the selection of time points from time series that contribute highly to the result and, thus, improve the models.

Our findings can be utilized in new data acquisition methods to control the data acquisition frequency. For instance, data acquisition could be increased during the interval of leaf and shoot development and less during the stage when the curd has reached 30% of the expected size, as plant development mainly takes place in the interior of the plant at this time. However, it is important to continuously observe the development from year to year and make adjustments as necessary, considering any variations in the development. To enhance generalization, it is imperative to collect additional data reflecting diverse weather and lighting conditions and additional data stemming from diverse developmental processes concerning the temporal occurrence of growth phases throughout the year, which can subsequently be assimilated into the model framework. Additionally, the findings in the application of cauliflower cultivation can be used to estimate the costs and benefits and determine whether the gain in accuracy justifies acquiring data weeks in advance. Our approach is adaptable and can be extended to other plant varieties or analogous time series analysis tasks.

Chapter 5

Overall Conclusion

This thesis presents significant advancements in the field of harvest-readiness prediction for cauliflower, addressing the critical need for accurate and non-invasive methods to determine optimal harvest times. The traditional challenges in cauliflower cultivation, such as the labor-intensive and subjective nature of manual harvesting, have been effectively tackled through innovative image-based approaches and advanced machine learning techniques. The foundation for this relies on the availability of open-source plant-specific datasets. Although there is a notable gap in the availability of these datasets, this work has made a significant contribution to reducing this gap by publishing an open-source image dataset, GrowliFlower, showing different developmental stages of georeferenced cauliflower plants. The experiments listed in this thesis demonstrate that image-based monitoring and analysis provide an efficient foundation for predicting cauliflower harvest-readiness. The provided dataset GrowliFlower is an initial step towards achieving this goal. However, due to the limited time period of the project in which this thesis is conducted, the dataset lacks diversity in terms of weather conditions, time intervals between acquisition dates, and plant varieties sown for an extended feature domain and ensured generalizability.

Our research demonstrates that integrating drone-based monitoring with deep learning models enables automated and precise predictions of harvest-readiness. Despite potential errors arising from field variability and limited training data, our comprehensive assessment and comparison of different models, like convolution-based Residual Networks and attention-based Vision Transformers, have provided valuable insights into forecasting times and prediction goals. The application of interpretable machine learning, particularly through the analysis of saliency maps, has allowed us to derive reliability scores for each classification result. This methodology informs farmers, enhancing their decision-making process and improving the accuracy of model predictions for unseen data.

The investigations of time series data for plant phenotyping have further contributed to our understanding of the decision-making process on harvest-readiness prediction, highlighting the importance of careful selection of acquisition days to increase model accuracy. The use of the interpretation technique GroupSHAP has identified key acquisition days and developmental stages that positively influence model performance, especially for our case. By focusing on these critical time points, we have demonstrated that it is possible to achieve significant improvements in accuracy while reducing the frequency of data collection, thus enhancing the efficiency of future agricultural practices.

Further insights gained during our work indicate that interpretable machine learning significantly enhances model performance, often surpassing the accuracy gains from hyperparameter tuning, which is time-intensive due to repeated retraining. Therefore, interpretable machine learning offers a more efficient approach to improving models.

Furthermore, it has been demonstrated that farmers' decisions and the economic situation significantly influence field workers' harvest decisions. Therefore, we recommend that the determination of harvest-readiness integrate both the visual assessment of the plants and economic factors.

5.1 Key Contributions

Our first contribution addresses the gap in the availability of plant-specific datasets by presenting GrowliFlower, an agricultural dataset focused on developing cauliflower plants. This dataset encompasses image time series of georeferenced cauliflower plants, coupled with phenological development data over time, derived from reference data. Additionally, it includes instance segmentation masks of plant components like plant instances, leaf instances or stem instances. This dataset facilitates the development of machine learning models for phenotyping traits analysis, enabling tasks such as harvest-readiness prediction, growth analysis, leaf counting, and more. GrowliFlower is crucial to this work, serving as the primary basis for all experimental investigations and analyses. It has been demonstrated that GrowliFlower provides a robust foundation for analyzing cauliflower's harvest-readiness. By creating and using GrowliFlower, additional challenges have been identified, which should be considered to improve approaches in future studies. Over several years of data collection, challenges have emerged due to various factors such as different cultivation varieties (varying degrees of self-coverage), the influence of field and plant handling by workers, and the impact of field size on harvest runs (whether the entire field is examined or not). The variation in these aspects particularly affects the generalizability of models and should be addressed in further data collection efforts.

Our second contribution diverges from previous methods that relied on statistical analyses of temperature and geometric properties. Instead, we utilize image-based analyses of harvest-readiness to investigate forecasting times and differentiate between two prediction tasks: binary classification of harvest-readiness and multi-class classification of harvest-readiness with respect to specific harvest days. For our use case, we demonstrate that even with a larger lead time, our approach achieves comparable accuracy to models using development stages closer to the harvest.

Our third contribution is the development of a framework for the reliability assessment of classification predictions. We calculate a reliability score using interpretable and unsupervised machine learning techniques that can be used to disseminate the reliability of predictions to the end user and adjust model predictions accordingly. This framework operates post-hoc during inference time and does not require human interaction, ensuring seamless integration into existing workflows.

Our fourth contribution involves the analysis of harvest-readiness prediction based on image time series data, integrating the developmental stages of cauliflower. We demonstrate that incorporating additional information improves the resulting model accuracies. This approach enhances predictive performance by leveraging the temporal progression of plant growth, thereby providing more accurate and reliable harvest-readiness predictions.

Our fifth contribution is the derivation of the acquisition time points in a time series and, consequently, the developmental stages of cauliflower plants at these time points, contributing to high model accuracy in predicting harvest-readiness. We utilize the interpretable machine learning method GroupSHAP for this purpose. The selective identification of developmental stages enables us to recommend reduced data collection for future datasets, assuming adaptations to varying growth patterns and climatic conditions.

5.2 Open Source Contributions

In the scope of this thesis, our dataset GrowliFlower is published open-source and can be found and downloaded on the PhenoRoam website:

- *https://phenoroam.phenorob.de/*

Based on the number of downloads and citations, it is evident that the dataset is being well-received within both the Machine Learning and plant science communities. This uptake suggests a favorable reception and indicates a high level of interest in the dataset. Such acceptance will likely stimulate further research efforts and applications and potentially foster novel innovations across both disciplines.

Chapter 6

What's next

Based on the knowledge and experience gathered during this thesis, we identify several challenges and open questions. Additionally, we propose ideas to address these challenges and provide solutions to the questions raised.

First, we have identified a lack of generalizability in the harvest-readiness prediction of cauliflower across different fields. Future data collection should focus on capturing diverse data within the cauliflower domain, including diverse temporal periods within a year, across multiple years, and encompassing various fields and varieties. Such an endeavor would go beyond the current scope of the project within which this thesis was conducted. The recordings' lighting conditions should be improved to reduce variance over time. Additionally, insights gained from time series analyses can be utilized to capture more targeted growth stages. We also recommend examining the entire field section for harvest-ready plants to minimize human bias in the reference data, if applicable in real economic implementation.

In addition to increased data collection, methodological approaches can also solve the challenges. One way to deal with the varying visual appearances of different fields and align them is to use domain transfer techniques. Style transfer Machine Learning (ML) offers several methods to transfer the style of one image to another without changing the latter's content. This can be achieved using techniques such as Generative Adversarial Networks (GANs) [230], [231] or diffusion models [232], [233]. Another approach to improve the accuracy of the models is the use of domain-specific pretrained models [234] on plant data, which also has the advantage of reducing the number of required labeled images.

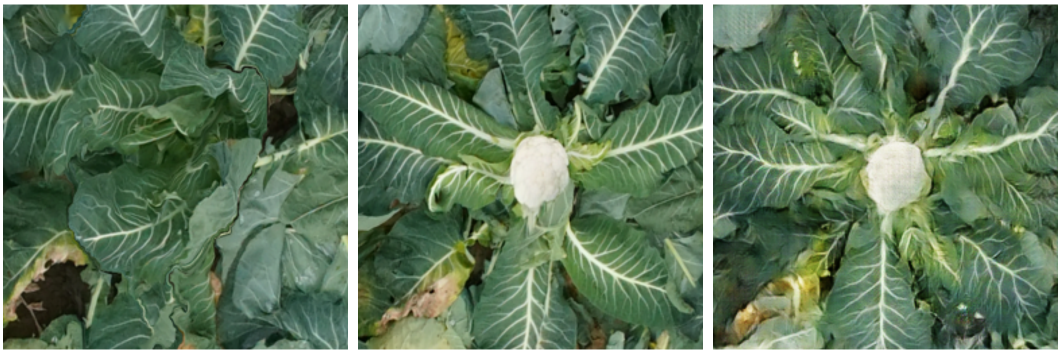
The high dependency of cauliflower growth on climate continues to pose challenges for accurately predicting harvest-readiness. Similarly, the significant economic influences on the harvest time point add to these challenges. Agricultural market dynamics influence the harvest decisions of farmers and their field workers and cannot be determined solely based on the visual appearance of the plants. Furthermore, maximizing yield does not always contribute to economic profit, as spatial competition with neighboring farmers must also be considered. Future research should focus on informed ML [235] and investigate whether integrating weather forecasting models and economic models can improve accuracy and foster generalizability.

Throughout our work, we have observed no direct neighborhood relationship regarding growth between the plants. Plants grow heterogeneously even when they are in close proximity. However, it has become evident during the work within this thesis that geographical location or neighborhood impacts growth. One cause could be, for example, soil conditions and nutrients that may be evenly distributed in the immediate area but unevenly over the whole field [236]. Based on the reference plots, we can see that plants within a single plot develop differently, but their development is more similar than plants from a different reference plot. Hence, we propose that further investigation into the correlation between geographical information and harvest-readiness be conducted in future studies and subsequently integrated into modeling frameworks.

So far, our work has focused on approaches using classification models to determine the harvest-readiness of cauliflower. However, this problem can also be addressed using regression models. The advantage of this approach is that regression models are able to predict time points between the designated harvest days. References are based on the harvest runs of field workers, not on the actual harvest-readiness, which can occur between the harvest runs.

Another direction in which future work can be advanced is the generation of unseen cauliflower curds and the subsequent derivation of geometric properties. For this, the defoliation dataset GrowliFlowerD can serve as a basis. We have already employed this approach in our work published by Kierdorf et al. [14], where we used GANs to generate a likely scenario behind the leaves to improve the estimation of the amount of harvest. In first experiments using a pix2pix GAN [104], we observe that we obtain sufficient results with the defoliated cauliflower data, as shown in Fig. 6.1, provided that (i) enough data are available, (ii) the center of the plant is not manually occluded by external cauliflower leaves, (iii) background and outer plant leaves stay similar between pre and post defoliation, and (iv) the resolution and contrast of the post-defoliation images are of high quality. Unfortunately, there is insufficient data for fields 1 and 2, the majority of images showing defoliated plants are blurry due to camera settings that failed to take into account the strong contrast between the white curd and the green plant. The example in Fig. 6.1 therefore represents a sample from the training set and may not be fully representative, but it demonstrates the potential of this approach.

In this thesis, we demonstrate that interpretable ML can be used to derive the reliability of predictions and the contribution of different acquisition time points



(a) Real (occluded). (b) Real (defoliated). (c) Generated (defoliated).

Figure 6.1: Example of generating the occluded curd under the canopy based on the GrowliFlowerD dataset. This example is generated using a pix2pix Generative Adversarial Network [104] and illustrates the real image showing the occluded cauliflower in (a), the real image showing the defoliated cauliflower in (b), and the generated image showing the defoliated plant.

and plant development states to model accuracy. Future work could expand the experiments using data-centric ML [237] to focus more closely on individual maps, investigating which features in the RGB space lead to unreliable predictions or negatively impact model accuracy. This approach may help to improve the foundational dataset by identifying which data samples exhibit higher uncertainty and understanding the underlying reasons.

Further work can investigate whether using multi-spectral data enhances the prediction of cauliflower harvest-readiness. Multi-spectral data provide insights into the physiological states of plants, such as their health status, nutrient content, and stress levels. Despite the farmer employing a robust stress management system, the plants are still affected by abiotic stresses such as intense sunlight, temperature fluctuations, and wind gusts. These factors may be reflected in the multi-spectral data and potentially influence harvest-readiness prediction. By incorporating multi-spectral imaging, researchers can more accurately assess how these abiotic stresses impact cauliflower development and may improve the precision of harvest-readiness predictions.

In the direction of time series analysis, a methodologically interesting approach could be the use of Graph Neural Networkss (GNNs) [238], which has proven to be promising in this and other areas [239]. GNNs provide a unique advantage when the time series data is represented as graphs, capturing the relationships and dependencies between different time points within the time series more effectively. Within the graph, a node could represent time points within the time series, and edges represent their temporal dependencies.

Before image-based harvest-readiness prediction using machine learning can be practically applied in the future, several of the aforementioned challenges must first be investigated and addressed.

Bibliography

[1] 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.

[2] P. Salter, "The growth and development of early summer cauliflower in relation to environmental factors," *Journal of horticultural Science*, vol. 35, no. 1, pp. 21–33, 1960.

[3] 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.

[4] J. Wheeler and P. Salter, "Effects of shortening the maturity period on harvesting costs of autumn cauliflowers," *Scientia Horticulturae*, vol. 2, no. 1, pp. 83–92, 1974.

[5] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[6] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sensing of Environment*, vol. 236, p. 111 402, 2020.

[7] 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.

[8] J. Kierdorf and R. Roscher, "Reliability scores from saliency map clusters for improved image-based harvest-readiness prediction in cauliflower," *IEEE Geoscience and Remote Sensing Letters*, 2023. DOI: 10.1109/LGRS.2023.3293802.

[9] J. Kierdorf, 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*, 2024. DOI: 10.3389/frai.2024.1416323.

[10] A. Emam, M. Farag, J. Kierdorf, L. Klingbeil, U. Rascher, and R. Roscher, "A framework for enhanced decision support in digital agriculture using explainable machine learning," *9th Workshop on Computer Vision in Plant Phenotyping and Agriculture (CVPPA)*, 2024. DOI: 10.13140/RG.2.2.24557.81121.

[11] N. Penzel, J. Kierdorf, R. Roscher, and J. Denzler, "Analyzing the behavior of cauliflower harvest-readiness models by investigating feature relevances," in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, IEEE, 2023, pp. 572–581. DOI: 10.1109/ICCVW60793.2023.00064.

[12] J. Kierdorf, J. Garcke, J. Behley, T. Cheeseman, and R. Roscher, "What identifies a whale by its fluke? On the benefit of interpretable machine learning for whale identification," *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 1005–1012, 2020. DOI: 10.5194/isprs-annals-V-2-2020-1005-2020.

[13] M. Farag, J. Kierdorf, and R. Roscher, "Inductive conformal prediction for harvest-readiness classification of cauliflower plants: A comparative study of uncertainty quantification methods," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 651–659. DOI: 10.1109/ICCVW60793.2023.00072.

[14] 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, ISSN: 2624-8212. DOI: 10.3389/frai.2022.830026.

[15] M. Günder, F. R. Ispizua Yamati, J. Kierdorf, R. Roscher, A.-K. Mahlein, and C. Bauckhage, "Agricultural plant cataloging and establishment of a data framework from uav-based crop images by computer vision," *GigaScience*, vol. 11, 2022. DOI: 10.1093/gigascience/giac054.

[16] 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," *Comput. Electron. Agric.*, vol. 190, p. 106 415, 2021, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2021.106415.

[17] D. Marcos, J. Kierdorf, T. Cheeseman, D. Tuia, and R. Roscher, “A whale’s tail-finding the right whale in an uncertain world,” in *xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, Springer, 2022, pp. 297–313. DOI: 10.1007/978-3-031-04083-2_15.

[18] Y. L. Chong, P. Nachtweide, J. Bauer, *et al.*, “Iterative AI-assisted annotation for visual pollinator identification and quantification in flower-enriched maize,” *Computers and Electronics in Agriculture*, 2024.

[19] P. Nachtweide, J. Bauer, J. Kierdorf, L. Chong, R. Roscher, R. Bayati, F. Kurth, T. F. Döring, A. Hamm, and S. J. Seidel, “Classical, audio-image- and video-based pollinator identification and quantification to evaluate biodiversity in agroecosystems,” in *64. Tagung der Gesellschaft für Pflanzenbauwissenschaften e. V. Digital tools, big data, modeling and sensing methods for sustainable and climate smart crop and grassland systems*, Göttingen, Germany, Oct. 2023.

[20] P. Nachtweide, J. Bauer, J. Kierdorf, L. Chong, R. Roscher, T. F. Döring, A. Hamm, and S. J. Seidel, “Classical and image-based pollinator identification and quantification in agroecosystems,” in *Fachforum Bienen und Landwirtschaft Strategiekonferenz*, Berlin, Germany, Jan. 2024.

[21] Y. Zhou, S. Booth, M. T. Ribeiro, and J. Shah, “Do feature attribution methods correctly attribute features?” In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 9623–9633.

[22] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, “How to explain individual classification decisions,” *The Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

[23] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, “Explaining explanations: An overview of interpretability of machine learning,” in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, IEEE, 2018, pp. 80–89.

[24] M. Saarela and S. Jauhiainen, “Comparison of feature importance measures as explanations for classification models,” *SN Applied Sciences*, vol. 3, pp. 1–12, 2021. DOI: 10.1007/s42452-021-04148-9.

[25] Y. Zhou, S. Booth, M. T. Ribeiro, and J. Shah, “Do feature attribution methods correctly attribute features?” In *Proceedings of the Innovations and Applications of Artificial Intelligence*, vol. 36, 2022, pp. 9623–9633.

[26] G. Casalicchio, C. Molnar, and B. Bischl, “Visualizing the feature importance for black box models,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, Springer, 2019, pp. 655–670.

[27] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley, “Explainable machine learning in deployment,” in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657. DOI: 10.1145/3351095.3375624.

[28] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Machine learning proceedings 1994*, Elsevier, 1994, pp. 121–129.

[29] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.

[30] E. Tjoa and C. Guan, “A survey on explainable artificial intelligence (xai): Toward medical xai,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[31] L. Weber, S. Lapuschkin, A. Binder, and W. Samek, “Beyond explaining: Opportunities and challenges of xai-based model improvement,” *Information Fusion*, 2022.

[32] E. Geisen and J. R. Bergstrom, *Usability testing for survey research.* Morgan Kaufmann, 2017.

[33] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, “Explainable artificial intelligence (xai) on timeseries data: A survey,” *arXiv preprint arXiv:2104.00950*, 2021.

[34] M. Yin, J. Wortman Vaughan, and H. Wallach, “Understanding the effect of accuracy on trust in machine learning models,” in *Proceedings of the 2019 chi conference on human factors in computing systems*, 2019, pp. 1–12.

[35] C. Berghoff, M. Neu, and A. von Twickel, “Vulnerabilities of connectionist ai applications: Evaluation and defense,” *Frontiers in big Data*, vol. 3, p. 23, 2020.

[36] X. Zhu, C. Vondrick, C. C. Fowlkes, and D. Ramanan, “Do we need more training data?” *International Journal of Computer Vision*, vol. 119, no. 1, pp. 76–92, 2016.

[37] Y. Chung, P. J. Haas, E. Upfal, and T. Kraska, “Unknown examples & machine learning model generalization,” *arXiv preprint arXiv:1808.08294*, 2018.

[38] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and S. Y. Philip, "Generalizing to unseen domains: A survey on domain generalization," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 8, pp. 8052–8072, 2022.

[39] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[40] R. Heale and A. Twycross, "Validity and reliability in quantitative studies," *Evidence-based nursing*, vol. 18, no. 3, pp. 66–67, 2015.

[41] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital signal processing*, vol. 73, pp. 1–15, 2018.

[42] M. Robnik-Šikonja and M. Bohanec, "Perturbation-based explanations of prediction models," *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, pp. 159–175, 2018.

[43] D. Doran, S. Schulz, and T. R. Besold, "What does explainable ai really mean? a new conceptualization of perspectives," *arXiv preprint arXiv:1710.00794*, 2017.

[44] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 8, pp. 2674–2693, 2018.

[45] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Comput. Vis. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 818–833, ISBN: 978-3-319-10590-1. DOI: 10.1007/978-3-319-10590-1_53.

[46] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[47] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[48] S. Thekdi and T. Aven, "Characterization of biases and their impact on the integrity of a risk study," *Safety science*, vol. 170, p. 106 376, 2024.

[49] R. Baeza-Yates, "Bias on the web," *Communications of the ACM*, vol. 61, no. 6, pp. 54–61, 2018.

[50] E. Sengupta, D. Garg, T. Choudhury, and A. Aggarwal, "Techniques to elimenate human bias in machine learning," in *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, 2018, pp. 226–230.

[51] T. J. Lark, I. H. Schelly, and H. K. Gibbs, "Accuracy, bias, and improvements in mapping crops and cropland across the united states using the usda cropland data layer," *Remote Sensing*, vol. 13, no. 5, p. 968, 2021.

[52] M. Christoph, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.

[53] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, pp. 457–506, 2021.

[54] R. Senge, S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier, "Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty," *Information Sciences*, vol. 255, pp. 16–29, 2014.

[55] C. Gruber, P. O. Schenk, M. Schierholz, F. Kreuter, and G. Kauermann, "Sources of uncertainty in machine learning–a statisticians' view," *arXiv preprint arXiv:2305.16703*, 2023.

[56] S. S. Afshari, F. Enayatollahi, X. Xu, and X. Liang, "Machine learning-based methods in structural reliability analysis: A review," *Reliability Engineering & System Safety*, vol. 219, p. 108 223, 2022.

[57] M. Basiri, *Context-aware data plausibility check using machine learning*, 2021.

[58] W. Jin, X. Li, and G. Hamarneh, "The xai alignment problem: Rethinking how should we evaluate human-centered ai explainability techniques," *arXiv preprint arXiv:2303.17707*, 2023.

[59] A. Jacovi and Y. Goldberg, "Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness?" *arXiv preprint arXiv:2004.03685*, 2020.

[60] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation," in *2018 IEEE symposium on security and privacy (SP)*, IEEE, 2018, pp. 3–18.

[61] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, pp. 1–30, 2019.

[62] B. Mucsányi, M. Kirchhof, E. Nguyen, A. Rubinstein, and S. J. Oh, "Trustworthy machine learning," *arXiv preprint arXiv:2310.08215*, 2023.

[63] X. Jin, L. Kumar, Z. Li, H. Feng, X. Xu, G. Yang, and J. Wang, "A review of data assimilation of remote sensing and crop models," *European journal of agronomy*, vol. 92, pp. 141–152, 2018.

[64] S. Hulbert and T. Orton, "Genetic and environmental effects on mean maturity date and uniformity in broccoli," *Journal of the American Society for Horticultural Science*, vol. 109, no. 4, pp. 487–490, 1984.

[65] H.-J. Wiebe, "Anbau von blumenkohl für eine kontinuierliche marktbelieferung während der erntesaison," *Gartenbauwissenschaft*, vol. 45, no. 6, pp. 282–288, 1980.

[66] R. Booij, "Cauliflower curd initiation and maturity: Variability within a crop," *Journal of Horticultural Science*, vol. 65, no. 2, pp. 167–175, 1990.

[67] K. Lindemann-Zutz, A. Fricke, and H. Stützel, "Prediction of time to harvest and its variability of broccoli (brassica oleracea var. italica) part ii. growth model description, parameterisation and field evaluation," *Scientia horticulturae*, vol. 200, pp. 151–160, 2016.

[68] V. Saiz-Rubio and F. Rovira-Más, "From smart farming towards agriculture 5.0: A review on crop data management," *Agronomy*, vol. 10, no. 2, p. 207, 2020.

[69] S. Sadik, "Morphology of the curd of cauliflower," *American Journal of Botany*, vol. 49, no. 3, pp. 290–297, 1962.

[70] K. Grevsen and J. E. Olesen, "Modelling cauliflower development from transplanting to curd initiation," *Journal of Horticultural Science*, vol. 69, no. 4, pp. 755–766, 1994.

[71] T. Wheeler, T. Hong, R. Ellis, G. Batts, J. Morison, and P. Hadley, "The duration and rate of grain growth, and harvest index, of wheat (triticum aestivum l.) in response to temperature and co2," *Journal of experimental botany*, vol. 47, no. 5, pp. 623–630, 1996.

[72] D. Wurr, J. R. Fellows, K. Phelps, and R. Reader, "Testing a vernalization model on field-grown crops of four cauliflower cultivars," *Journal of Horticultural Science*, vol. 69, no. 2, pp. 251–255, 1994.

[73] S. Pearson, P. Hadley, and A. Wheldon, "A model of the effects of temperature on the growth and development of cauliflower (brassica oleracea l. botrytis)," *Scientia Horticulturae*, vol. 59, no. 2, pp. 91–106, 1994.

[74] H. U. Rahman, P. Hadley, and S. Pearson, “Relationship between temperature and cauliflower (brassica oleracea l. var. botrytis) growth and development after curd initiation,” *Plant growth regulation*, vol. 52, pp. 61–72, 2007.

[75] M. Lohachov, R. Korei, K. Oki, K. Yoshida, I. Azechi, S. I. Salem, and N. Utsumi, “Rnn-based approach for broccoli harvest time forecast,” *Agronomy*, vol. 14, no. 2, p. 361, 2024.

[76] D. Wurr, J. R. Fellows, R. Sutherland, and E. Elphinstone, “A model of cauliflower curd growth to predict when curds reach a specified size,” *Journal of Horticultural Science*, vol. 65, no. 5, pp. 555–564, 1990.

[77] A. Rosen, Y. Hasan, W. Briggs, and R. Uptmoor, “Genome-based prediction of time to curd induction in cauliflower,” *Frontiers in Plant Science*, vol. 9, p. 256 209, 2018.

[78] G. Grenzdörffer, “Automatic generation of geometric parameters of individual cauliflower plants for rapid phenotyping using drone images,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 329–335, 2019.

[79] L. García-Pérez, J. Marchant, T. Hague, and M. García-Alegre, “Fuzzy decision system for threshold selection to cluster cauliflower plant blobs from field visual images,” *SCI2000, Orlando*, pp. 23–28, 2000.

[80] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, “Automatic crop detection under field conditions using the hsv colour space and morphological operations,” *Computers and electronics in agriculture*, vol. 133, pp. 97–107, 2017.

[81] 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.

[82] A. García-Manso, R. Gallardo-Caballero, C. J. García-Orellana, H. M. González-Velasco, and M. Macías-Macías, “Towards selective and automatic harvesting of broccoli for agri-food industry,” *Computers and Electronics in Agriculture*, vol. 188, p. 106 263, 2021.

[83] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote sensing and image interpretation*, Seventh. John Wiley & Sons, Feb. 2015, 736 pp., ISBN: 978-1118343289.

[84] E. R. Hunt Jr and C. S. Daughtry, “What good are unmanned aircraft systems for agricultural remote sensing and precision agriculture?” *International journal of remote sensing*, vol. 39, no. 15-16, pp. 5345–5376, 2018.

[85] W. H. Maes and K. Steppe, "Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture," *Trends in plant science*, vol. 24, no. 2, pp. 152–164, 2019.

[86] C. A. Nock, R. J. Vogt, and B. E. Beisner, "Functional traits," *eLS*, pp. 1–8, 2016. DOI: 10.1002/9780470015902.a0026282.

[87] D. Chaparro, M. Piles, M. Vall-Llossera, A. Camps, A. G. Konings, and D. Entekhabi, "L-band vegetation optical depth seasonal metrics for crop yield assessment," *Remote Sensing of Environment*, vol. 212, pp. 249–259, 2018.

[88] M. K. Mosleh, Q. K. Hassan, and E. H. Chowdhury, "Application of remote sensors in mapping rice area and forecasting its production: A review," *Sensors*, vol. 15, no. 1, pp. 769–791, 2015.

[89] A. Verger, F. Baret, and M. Weiss, "Near real-time vegetation monitoring at global scale," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 8, pp. 3473–3481, 2014.

[90] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geoscience Frontiers*, vol. 7, no. 1, pp. 3–10, 2016.

[91] M. Debolini, J. M. Schoorl, A. Temme, M. Galli, and E. Bonari, "Changes in agricultural land use affecting future soil redistribution patterns: A case study in southern tuscany (italy)," *Land Degradation & Development*, vol. 26, no. 6, pp. 574–586, 2015.

[92] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, *et al.*, "Deep learning and process understanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, 2019.

[93] I. Ali, F. Greifeneder, J. Stamenkovic, M. Neumann, and C. Notarnicola, "Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data," *Remote Sensing*, vol. 7, no. 12, pp. 16 398–16 421, 2015.

[94] J. Verrelst, Z. Malenovskỳ, C. Van der Tol, G. Camps-Valls, J.-P. Gastellu-Etchegorry, P. Lewis, P. North, and J. Moreno, "Quantifying vegetation biophysical variables from imaging spectroscopy data: A review on retrieval methods," *Surveys in Geophysics*, vol. 40, no. 3, pp. 589–629, 2019.

[95] M. Rußwurm and M. Korner, "Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[96] M. Rußwurm and M. Körner, "Self-attention for raw optical satellite time series classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 169, pp. 421–435, 2020.

[97] C. Pelletier, G. I. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sensing*, vol. 11, no. 5, p. 523, 2019, https://www.mdpi.com/2072-4292/11/5/523.

[98] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *European conference on computer vision*, Springer, 2016, pp. 312–329.

[99] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6656–6664.

[100] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, *et al.*, "Leaf segmentation in plant phenotyping: A collation study," *Machine vision and applications*, vol. 27, no. 4, pp. 585–606, 2016.

[101] J. Weyler, A. Milioto, T. Falck, J. Behley, and C. Stachniss, "Joint plant instance detection and leaf count estimation for in-field plant phenotyping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3599–3606, 2021.

[102] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *sensors*, vol. 16, no. 8, p. 1222, 2016.

[103] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, *et al.*, "Development of a sweet pepper harvesting robot," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1027–1039, 2020.

[104] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[105] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[106] P. Lottes, M. Hörferlin, S. Sander, and C. Stachniss, "Effective vision-based classification for separating sugar beets and weeds for precision farming," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1160–1178, 2017.

[107] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, "Fully convolutional networks with sequential information for robust crop and weed detection in precision farming," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2870–2877, 2018.

[108] A. Milioto, P. Lottes, and C. Stachniss, “Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns,” in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 2229–2235.

[109] A. Ahmadi, M. Halstead, and C. McCool, “Virtual temporal samples for recurrent neural networks: Applied to semantic segmentation in agriculture,” *arXiv preprint arXiv:2106.10118*, 2021.

[110] S. Mostafa, D. Mondal, K. Panjvani, L. Kochian, and I. Stavness, “Explainable deep learning in plant phenotyping,” *Frontiers in Artificial Intelligence*, vol. 6, p. 1 203 546, 2023.

[111] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in plant science*, vol. 7, p. 215 232, 2016.

[112] C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson, and H. Lipson, “Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning,” *Phytopathology*, vol. 107, no. 11, pp. 1426–1432, 2017.

[113] A. Dobrescu, M. Valerio Giuffrida, and S. A. Tsaftaris, “Leveraging multiple datasets for deep leaf counting,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 2072–2079.

[114] G. Bernotas, L. C. Scorza, M. F. Hansen, I. J. Hales, K. J. Halliday, L. N. Smith, M. L. Smith, and A. J. McCormick, “A photometric stereo-based 3d imaging system using computer vision and deep learning for tracking plant growth,” *GigaScience*, vol. 8, no. 5, giz056, 2019.

[115] 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, p. 105 117, 2020.

[116] R. Sapkota, D. Ahmed, and M. Karkee, “Creating image datasets in agricultural environments using dall. e: Generative ai-powered large language model,” *Qeios*, 2024.

[117] T. Akagi, M. Onishi, K. Masuda, R. Kuroki, K. Baba, K. Takeshita, T. Suzuki, T. Niikawa, S. Uchida, and T. Ise, “Explainable deep learning reproduces a ‘professional eye’on the diagnosis of internal disorders in persimmon fruit,” *Plant and Cell Physiology*, vol. 61, no. 11, pp. 1967–1973, 2020.

[118] K. Wei, B. Chen, J. Zhang, S. Fan, K. Wu, G. Liu, and D. Chen, “Explainable deep learning study for leaf disease classification,” *Agronomy*, vol. 12, no. 5, p. 1035, 2022.

[119] Y. Toda and F. Okura, “How convolutional neural networks diagnose plant disease,” *Plant Phenomics*, 2019.

[120] S. Kinger and V. Kulkarni, “Explainable ai for deep learning based disease detection,” in *2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)*, 2021, pp. 209–216.

[121] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[122] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?” explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on KDD*, 2016, pp. 1135–1144.

[123] A. Mateo-Sanchis, J. E. Adsuara, M. Piles, J. Munoz-Marí, A. Perez-Suay, and G. Camps-Valls, “Interpretable long short-term memory networks for crop yield estimation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[124] H. Chandra, P. M. Pawar, R. Elakkiya, P. S. Tamizharasan, R. Muthalagu, and A. Panthakkan, “Explainable ai for soil fertility prediction,” *IEEE Access*, 2023.

[125] M. Y. Shams, S. A. Gamel, and F. M. Talaat, “Enhancing crop recommendation systems with explainable artificial intelligence: A study on agricultural decision-making,” *Neural Computing and Applications*, vol. 36, no. 11, pp. 5695–5714, 2024.

[126] M. Gill, R. Anderson, H. Hu, M. Bennamoun, J. Petereit, B. Valliyodan, H. T. Nguyen, J. Batley, P. E. Bayer, and D. Edwards, “Machine learning models outperform deep learning models, provide interpretation and facilitate feature selection for soybean trait prediction,” *BMC plant biology*, vol. 22, no. 1, p. 180, 2022.

[127] S. Majumder and C. M. Mason, “A machine learning approach to study plant functional trait divergence,” *Applications in Plant Sciences*, e11576, 2023.

[128] D. Paudel, A. de Wit, H. Boogaard, D. Marcos, S. Osinga, and I. N. Athanasiadis, “Interpretability of deep learning models for crop yield forecasting,” *Computers and Electronics in Agriculture*, vol. 206, p. 107 663, 2023.

[129] A. Dobrescu, M. Valerio Giuffrida, and S. A. Tsaftaris, "Understanding deep neural networks for regression in leaf counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[130] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, "Deep learning for plant identification using vein morphological patterns," *Computers and electronics in agriculture*, vol. 127, pp. 418–424, 2016.

[131] L. Drees, I. Weber, M. Rußwurm, and R. Roscher, "Time dependent image generation of plants from incomplete sequences with cnn-transformer," in *DAGM German Conference on Pattern Recognition*, Springer, 2022, pp. 495–510. DOI: 10.1007/978-3-031-16788-1_30.

[132] S. Kolhar and J. Jagtap, "Spatio-temporal deep neural networks for accession classification of arabidopsis plants using image sequences," *Ecological Informatics*, vol. 64, p. 101 334, 2021. DOI: 10.1016/j.ecoinf.2021.101334.

[133] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of xai methods on time series," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, 2019, pp. 4197–4201. DOI: 10.1109/ICCVW.2019.00516.

[134] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, "Explainable ai for time series classification: A review, taxonomy and research directions," *IEEE Access*, vol. 10, pp. 100 700–100 724, 2022. DOI: 10.1109/ACCESS.2022.3207765.

[135] M. Villani, J. Lockhart, and D. Magazzeni, "Feature importance for time series data: Improving kernelshap," *arXiv preprint arXiv:2210.02176*, 2022. DOI: 10.48550/arXiv.2210.02176.

[136] B. Shickel and P. Rashidi, "Sequential interpretability: Methods, applications, and future direction for understanding deep learning models in the context of sequential data," *arXiv preprint arXiv:2004.12524*, 2020. DOI: 10.48550/arXiv.2004.12524.

[137] S. A. Siddiqui, D. Mercier, M. Munir, A. Dengel, and S. Ahmed, "Tsviz: Demystification of deep learning models for time-series analysis," *IEEE Access*, vol. 7, pp. 67 027–67 040, 2019. DOI: 10.1109/ACCESS.2019.2912823.

[138] R. Leygonie, S. Lobry, and L. Wendling, "An a contrario approach for plant disease detection," in *BMVC Workshop*, 2023.

[139] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[140] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[141] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[142] M. F. Minamikawa, K. Nonaka, H. Hamada, T. Shimizu, and H. Iwata, "Dissecting breeders' sense via explainable machine learning approach: Application to fruit peelability and hardness in citrus," *Frontiers in Plant Science*, vol. 13, p. 832 749, 2022.

[143] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting, "Making deep neural networks right for the right scientific reasons by interacting with their explanations," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 476–486, 2020.

[144] T. Kim, H. Kim, K. Baik, and Y. Choi, "Instance-aware plant disease detection by utilizing saliency map and self-supervised pre-training," *Agriculture*, vol. 12, no. 8, 2022, ISSN: 2077-0472. DOI: `10.3390/agriculture12081084`.

[145] V. Pendyala and H. Kim, "Assessing the reliability of machine learning models applied to the mental health domain using explainable ai," *Electronics*, vol. 13, no. 6, p. 1025, 2024.

[146] B. Kailkhura, B. Gallagher, S. Kim, A. Hiszpanski, and T. Y.-J. Han, "Reliable and explainable machine-learning methods for accelerated material discovery," *npj Computational Materials*, vol. 5, no. 1, p. 108, 2019.

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

[148] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[149] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on the Future of Datasets in Vision*, vol. 2, 2015.

[150] N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss, "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1045–1052, 2017.

[151] A. Förster, J. Behley, J. Behmann, and R. Roscher, "Hyperspectral plant disease forecasting using generative adversarial networks," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, pp. 1793–1796.

[152] J. Kierdorf, L. Zabawa, L. Lucks, L. Klingbeil, H. Kuhlmann, *et al.*, "Detection and counting of wheat ears by means of ground-based image acquisition.," *Bornimer Agrartechnische Berichte*, vol. 1, no. 102, pp. 158–167, 2019.

[153] L. Zabawa, A. Kicherer, L. Klingbeil, A. Milioto, R. Topfer, H. Kuhlmann, and R. Roscher, "Detection of single grapevine berries in images using fully convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[154] M. Halstead, S. Denman, C. Fookes, and C. McCool, "Fruit detection in the wild: The impact of varying conditions and cultivar," in *2020 Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 2020, pp. 1–8.

[155] H. Scharr, M. Minervini, A. Fischbach, and S. A. Tsaftaris, "Annotated image datasets of rosette plants," in *European Conference on Computer Vision. Zürich, Suisse*, 2014, pp. 6–12.

[156] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern recognition letters*, vol. 81, pp. 80–89, 2016.

[157] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *arXiv preprint arXiv:1712.00580*, 2017.

[158] D. Ward and P. Moghadam, "Synthetic arabidopsis dataset," in *CSIRO. Data Collection.*, 2018.

[159] K. Kusumam, T. Krajník, S. Pearson, T. Duckett, and G. Cielniak, "3d-vision based detection, localization, and sizing of broccoli heads in the field," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1505–1518, 2017. DOI: https://doi.org/10.1002/rob.21726. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21726.

[160] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak, "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture," *Journal of Field Robotics*, vol. 37, no. 1, pp. 7–19, 2020.

[161] P. M. Blok, E. J. van Henten, F. K. van Evert, and G. Kootstra, "Image-based size estimation of broccoli heads under varying degrees of occlusion," *biosystems engineering*, vol. 208, pp. 213–233, 2021. DOI: https://doi.org/10.4121/13603787.v1.

[162] J. Weyler, F. Magistri, E. Marks, Y. L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley, "Phenobench–a large dataset and benchmarks for semantic image interpretation in the agricultural domain," *arXiv preprint arXiv:2306.04557*, 2023.

[163] A. Ahmadi, M. Halstead, and C. McCool, "Towards autonomous crop-agnostic visual navigation in arable fields," *CoRR*, 2021.

[164] U. Sara, A. Rajbongshi, R. Shakil, B. Akter, and M. S. Uddin, "Vegnet: An organized dataset of cauliflower disease for a sustainable agro-based automation system," *Data in Brief*, vol. 43, p. 108 422, 2022.

[165] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[166] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[167] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[168] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[169] C. M. Bishop, *Pattern recognition and machine learning.* springer, 2006.

[170] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[171] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[172] C. Hennig, M. Meila, F. Murtagh, and R. Rocci, "Spectral clustering: A tutorial for the 2010's," in *Handbook of cluster analysis*, CRC Press, 2016, pp. 1–23.

[173] A. Y. Ng, M. I. Jordan, Y. Weiss, *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[174] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[175] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[176] A. B. Tsybakov, *Introduction to Nonparametric Estimation.* Springer, 2008.

[177] 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.

[178] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[179] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[180] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[181] P. A. Marin Zapata, S. Roth, D. Schmutzler, T. Wolf, E. Manesso, and D.-A. Clevert, "Self-supervised feature extraction from image time series in plant phenotyping using triplet networks," *Bioinformatics*, vol. 37, no. 6, pp. 861–867, 2021.

[182] 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 *Proceedings of the IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 325–12 334. DOI: 10.1109/CVPR42600.2020.01234.

[183] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[184] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*, PMLR, 2017, pp. 1243–1252.

[185] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2.

[186] 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 *European conference on computer vision*, Springer, 2014, pp. 740–755.

[187] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar, 2020.

[188] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav),” in *International conference on machine learning*, PMLR, 2018, pp. 2668–2677.

[189] M. Ibrahim, M. Louie, C. Modarres, and J. Paisley, “Global explanations of neural networks: Mapping the landscape of predictions,” in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 279–287.

[190] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, “Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model,” *The Annals of Applied Statistics*, vol. 9, no. 3, Sep. 2015, ISSN: 1932-6157. DOI: 10.1214/15-aoas848.

[191] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1721–1730.

[192] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.

[193] Z. C. Lipton, “The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.,” *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[194] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, “Visualizing the loss landscape of neural nets,” *Advances in neural information processing systems*, vol. 31, 2018.

[195] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, “Unmasking clever hans predictors and assessing what machines really learn,” *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[196] E. Soares, P. P. Angelov, B. Costa, M. P. G. Castro, S. Nageshrao, and D. Filev, “Explaining deep learning models through rule-based approximation and visualization,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 8, pp. 2399–2407, 2020.

[197] B. Kim, R. Khanna, and O. O. Koyejo, “Examples are not enough, learn to criticize! criticism for interpretability,” *Advances in neural information processing systems*, vol. 29, 2016.

[198] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *International Conference on Machine Learning*, PMLR, 2017, pp. 1885–1894.

[199] A. Ghorbani, A. Abid, and J. Zou, “Interpretation of neural networks is fragile,” in *Proceedings of the Innovations and Applications of Artificial Intelligence*, vol. 33, 2019, pp. 3681–3688.

[200] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *International conference on machine learning*, PMLR, 2017, pp. 3145–3153.

[201] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, “Layer-wise relevance propagation: An overview,” *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[202] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *International conference on machine learning*, PMLR, 2017, pp. 3319–3328.

[203] A. Maniaci, P. M. Riela, G. Iannella, J. R. Lechien, I. La Mantia, M. De Vincentiis, G. Cammaroto, C. Calvo-Henriquez, M. Di Luca, C. Chiesa Estomba, *et al.*, “Machine learning identification of obstructive sleep apnea severity through the patient clinical features: A retrospective study,” *Life*, vol. 13, no. 3, p. 702, 2023.

[204] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” *arXiv preprint arXiv:1412.6806*, 2014.

[205] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, “Towards better understanding of gradient-based attribution methods for deep neural networks,” *arXiv preprint arXiv:1711.06104*, 2017.

[206] C. Burns, J. Thomason, and W. Tansey, “Interpreting black box models via hypothesis testing,” in *Proceedings of the 2020 ACM-IMS on foundations of data science conference*, 2020, pp. 47–57.

[207] G. Erion, J. D. Janizek, P. Sturmfels, S. Lundberg, and S.-I. Lee, *Improving performance of deep learning models with axiomatic attribution priors and expected gradients*, 2020. arXiv: 1906.10670 [cs.LG].

[208] D. Alvarez-Melis and T. S. Jaakkola, “On the robustness of interpretability methods,” *arXiv preprint arXiv:1806.08049*, 2018.

[209] M. Jullum, A. Redelmeier, and K. Aas, "Groupshapley: Efficient prediction explanation with shapley values for feature groups," *arXiv preprint arXiv:2106.12228*, 2021.

[210] M. Persia, E. Barca, R. Greco, M. I. Marzulli, and P. Tartarino, "Archival aerial images georeferencing: A geostatistically-based approach for improving orthophoto accuracy with minimal number of ground control points," *Remote Sensing*, vol. 12, no. 14, p. 2232, 2020. DOI: `10.3390/rs12142232`.

[211] C. Feller, H. Bleiholder, L. Buhr, H. Hack, M. Hess, R. Klose, U. Meier, R. Stauss, T. Boom, and E. Weber, "Phenological growth stages of vegetable crops. ii. fruit vegetables and pulses. coding and description according to the extended bbch scale with illustrations," *Nachrichtenblatt des Deutschen Pflanzenschutzdienstes*, vol. 47, no. 9, pp. 217–232, 1995.

[212] H.-J. Wiebe, "Effect of temperature and light on the growth and development of cauliflowers. 1. the duration of the seedling stage for vernalization.," *Gartenbauwissenschaft*, vol. 37, no. 3, pp. 165–178, 1972.

[213] H.-J. Wiebe, "Effect of temperature and light on the growth and development of cauliflower. ii. optimum vernalization temperature and period.," *Gartenbauwissenschaft*, vol. 37, no. 4, pp. 293–303, 1972.

[214] D. Wurr, J. M. Akehurst, and T. Thomas, "A hypothesis to explain the relationship between low-temperature treatment, gibberellin activity, curd initiation and maturity of cauliflower," *Scientia Horticulturae*, vol. 15, no. 4, pp. 321–330, 1981.

[215] J. Williams, C. Jones, J. Kiniry, and D. A. Spanel, "The epic crop growth model," *Transactions of the ASAE*, vol. 32, no. 2, pp. 497–0511, 1989.

[216] J. Weyler, F. Magistri, P. Seitz, J. Behley, and C. Stachniss, "In-field phenotyping based on crop leaf and plant instance segmentation," in *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2022.

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

[218] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: Detection and saliency map visualisation," in *Human and machine learning*, Springer, 2018, pp. 93–117.

[219] M. O. Turkoglu, S. D'Aronco, G. Perich, F. Liebisch, C. Streit, K. Schindler, and J. D. Wegner, "Crop mapping from image time series: Deep learning with multi-scale label hierarchies," *Remote Sensing of Environment*, vol. 264, p. 112 603, 2021. DOI: `10.1016/j.rse.2021.112603`.

[220] M. Rußwurm, N. Courty, R. Emonet, S. Lefèvre, D. Tuia, and R. Tavenard, "End-to-end learned early classification of time series for in-season crop type mapping," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 445–456, 2023. DOI: 10.1016/j.isprsjprs.2022.12.016.

[221] 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.

[222] B. Schauberger, J. Jägermeyr, and C. Gornott, "A systematic review of local to regional yield forecasting approaches and frequently used data resources," *European Journal of Agronomy*, vol. 120, p. 126 153, 2020. DOI: 10.1016/j.eja.2020.126153.

[223] M. Yli-Heikkilä, S. Wittke, M. Luotamo, E. Puttonen, M. Sulkava, P. Pellikka, J. Heiskanen, and A. Klami, "Scalable crop yield prediction with sentinel-2 time series and temporal convolutional network," *Remote Sensing*, vol. 14, no. 17, p. 4193, 2022. DOI: 10.3390/rs14174193.

[224] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, "“everyone wants to do the model work, not the data work”: Data cascades in high-stakes ai," in *Proceedings of the 2021 ACM CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15. DOI: 10.1145/3411764.3445518.

[225] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*, IEEE, 2016, pp. 1–6. DOI: 10.1109/QoMEX.2016.7498955.

[226] M. Tollenaar, J. Fridgen, P. Tyagi, P. W. Stackhouse Jr, and S. Kumudini, "The contribution of solar brightening to the us maize yield trend," *Nature Climate Change*, vol. 7, no. 4, pp. 275–278, 2017. DOI: 10.1038/nclimate3234.

[227] C. V. Bratu, T. Muresan, and R. Potolea, "Improving classification accuracy through feature selection," in *2008 4th International Conference on Intelligent Computer Communication and Processing*, IEEE, 2008, pp. 25–32. DOI: 10.1109/ICCP.2008.4648350.

[228] C. Chu, A.-L. Hsu, K.-H. Chou, P. Bandettini, C. Lin, A. D. N. Initiative, *et al.*, "Does feature selection improve classification accuracy? impact of sample size and feature selection on classification using anatomical magnetic resonance images," *Neuroimage*, vol. 60, no. 1, pp. 59–70, 2012. DOI: 10.1016/j.neuroimage.2011.11.066.

[229] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosc. Remote Send. Lett*, vol. 12, no. 11, pp. 2321–2325, 2015. DOI: 10.1109/LGRS.2015.2475299.

[230] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[231] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[232] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, and C. Xu, "Inversion-based style transfer with diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 10 146–10 156.

[233] Z. Wang, L. Zhao, and W. Xing, "Stylediffusion: Controllable disentangled style transfer via diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7677–7689.

[234] G. Roggiolani, F. Magistri, T. Guadagnino, J. Weyler, G. Grisetti, C. Stachniss, and J. Behley, "On domain-specific pre-training for effective semantic perception in agricultural robotics," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11 786–11 793.

[235] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, *et al.*, "Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 614–633, 2021.

[236] R. N. Roy *et al.*, *Plant nutrition for food security. A guide for integrated nutrient management.* FAO Fertilizer and Plant Nutrition Bulletin 16, 2006, pp. 201–214, FAO (Food and Agriculture Organization of the United Nations).

[237] 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.

[238] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[239] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, “A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection,” *arXiv preprint arXiv:2307.03759*, 2023.

Acronyms

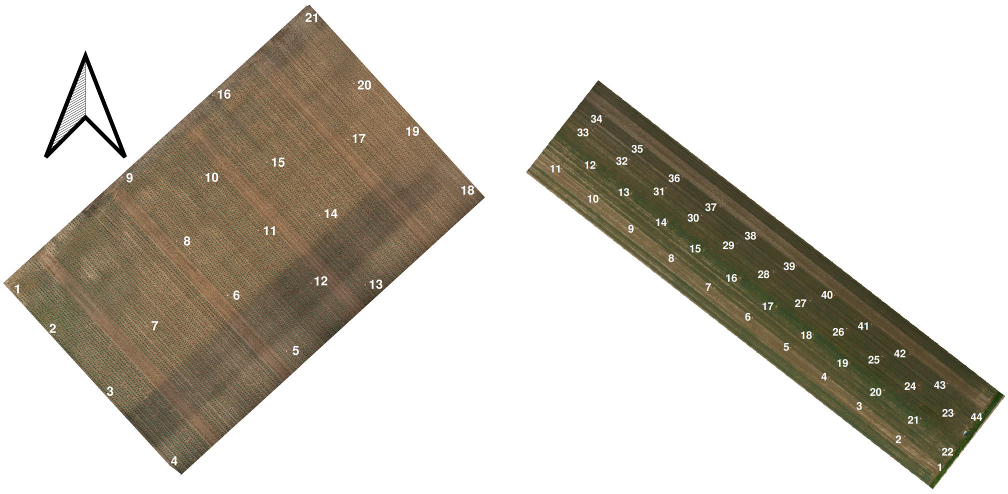
bcAcc	balanced class accuracy
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GNN	Graph Neural Networks
DAP	day after planting
DL	Deep Learning
EM	Expectation-Maximization
Grad-CAM	Gradient-weighted Class Activation Mapping
HD	harvest day
iTS	initial time series
IoU	Intersection over Union
KDE	Kernel Density Estimation
kNN	k-Nearest-Neighbors
LIME	Local Interpretable Model-agnostic Explanations
ML	Machine Learning
MLP	multi-layer perceptron
NLP	Natural Language Processing
NN	neural network
OSM	Occlusion Sensitivity Mapping
oaAcc	overall accuracy
ResNet	Residual Network
PE	positional encoding
SC	Spectral Clustering
SHAP	Shapley Additive Explanations

sTS	selective time series
TE	temporal encoding
TPE	time point embedding
TSE	time series embedding
ViT	Vision Transformer

List of Figures

List of Tables

Appendices



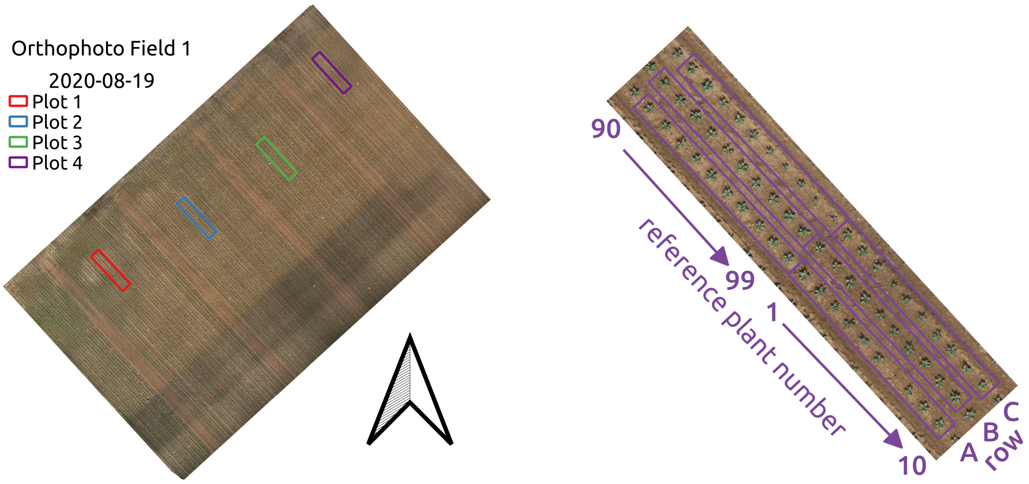
(a) Field 1

(b) Field 2

Figure 2: Location of GCPs in fields 1 and 2. Figure source: Kierdorf et al. [7].

Table 1: Monitored abiotic and biotic stresses.

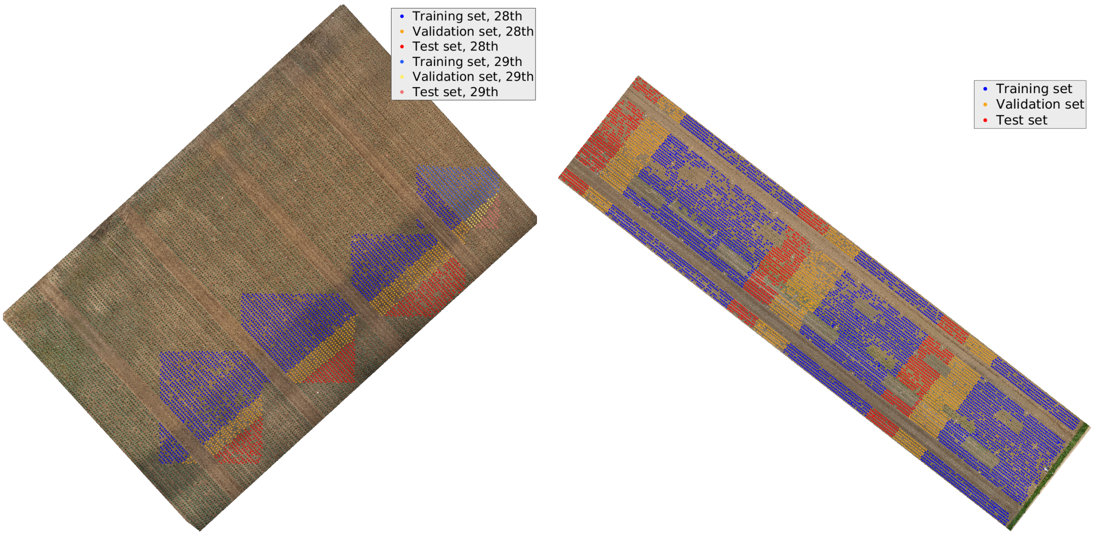
Abbreviation	Meaning	Abbreviation	Meaning
P	plant	L	leaf/leaves
nP	no plant	wL	without leaves
Pl	plant lying down	oL	old leaves
wP	whole plant	yL	yellowish leaves
2P	2 plants	rL	reddish leaves
bb	blind bud	pgL	pale green leaves
pd	planted too deep	pygL	pale yellowish green leaves
A	aphids present	sg	stunted growth with many shoots
C	coal fleas present	dT	damage to leaves caused by tractor
F	flies present		



(a)

(b)

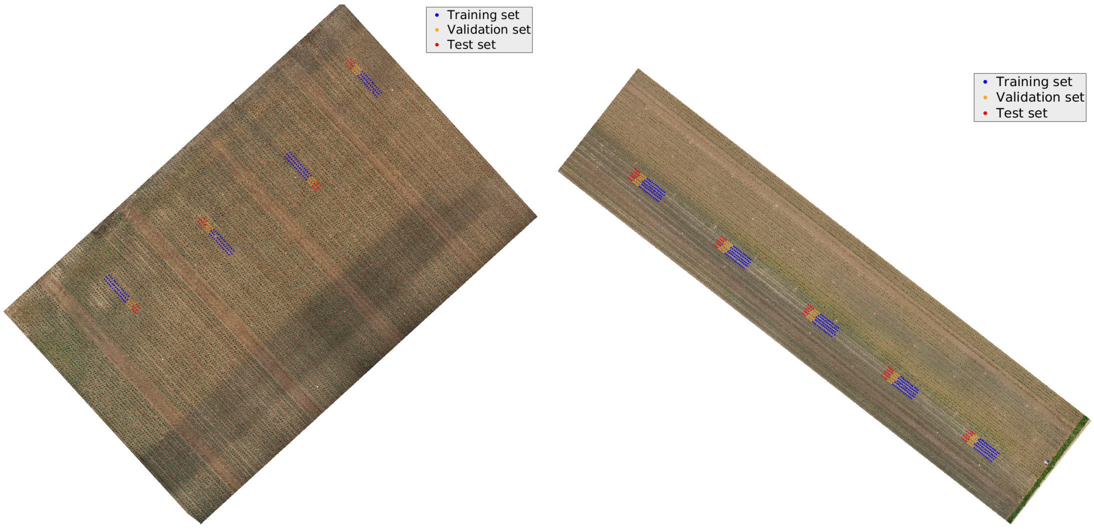
Figure 3: Visual overview of (a) reference plots for in-situ measurements in field 1 and (b) the respective design of reference plot 4 (including reference plants and ordering of reference plant numbers). The plot design is valid for all reference plots in field 1. Figure source: Kierdorf et al. [7].



(a) Field 1

(b) Field 2

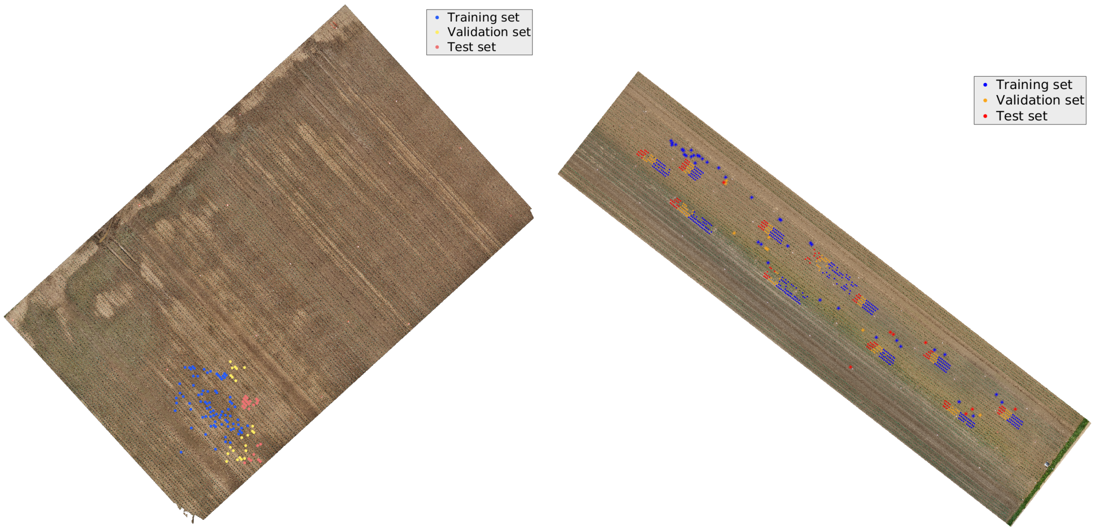
Figure 4: Separation of plants within fields 1 and 2 in GrowliFlowerT in training (blue), validation (yellow), and testing (red) sets. For field 1, the two planting days are separated using dark colors for July, 28th, 2020 and light colors for July, 29th, 2020. Figure source: Kierdorf et al. [7].



(a) Field 1

(b) Field 2

Figure 5: Separation of reference plants in fields 1 and 2 in GrowliFlowerR in training (blue), validation (yellow), and testing (red) sets. Figure source: Kierdorf et al. [7].



(a) Field 1

(b) Field 2

Figure 6: Separation of defoliated plants in fields 1 and 2 in GrowliFlowerD in training (blue), validation (yellow), and testing (red) sets. Figure source: Kierdorf et al. [7].