Generative Models for Image-Based Plant Analysis: Thermal Image Super-Resolution and Plant Architecture Generation

By

HEESUP YUN

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biological Systems Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

J. Mason Earles, Chair

Brian N. Bailey

Christine H. Diepenbrock

Committee in Charge

2026

To my loving wife, Gayoung

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Mason Earles, for his invaluable guidance and unwavering support, and for his mentorship and feedback, which helped me become an independent researcher. I regard him as an academic, professional, and personal role model.

I thank my dissertation committee for their guidance throughout this research and dissertation. I thank Dr. Brian N. Bailey for developing the Helios plant simulation library, which was central to my research, for generously helping me use the program, and for sharing his insightful ideas. I thank Dr. Christine Diepenbrock for teaching me the fundamentals of plant breeding, leading the field experiments that enabled me to collect essential data, and emphasizing the importance of statistical analysis. I thank my Qualifying Exam committee members—Dr. David C. Slaughter, Dr. Stavros G. Vougioukas, and Dr. Alireza Pourreza—for guiding me through that milestone.

This research and dissertation were completed with support from the G×E×M Innovation in Intelligence for Climate Adaptation project. I gratefully acknowledge the Gates Foundation for funding this project. I thank Betsy Huynh for her administrative assistance throughout the project. I thank the members of the Plant Simulation Laboratory—Dr. Ismael Mayanja, Dr. Tong Lei, and Dr. Ioannis Droutsas—and the Diepenbrock Lab—Dr. Sassoum Lo, Jonathan Berlingeri, and Margaret Riggs—for their collaboration. My colleagues from the AI-Enabled Sensing team—Earl Ranario, Summer Reeves, and Riya Desai—deserve special recognition for their tireless efforts in data collection while traveling from Davis to Kearney in 2022 and 2023. I thank Lars Lundqvist for his dedication to helping me build the mobile weather station and collect data in Davis. I acknowledge the Google X Mineral and Farm-ng teams for providing rover hardware and technical support.

I thank my past and current colleagues in the Plant AI and Biophysics Lab—Dr. Alexander Olenskyj, Dr. Hamid Kamangir, Dr. Darío Guevara, Dr. Andres Montes de Oca, Dr. Soichiro Nishiyama, Dr. Afef Marzougui, Apoorva Jha, and Vishal Singh—for the memories and friendships we built together. I am especially indebted to Dr. Vivian Vuong, who guided me through the pre-admission orientation and Qualifying Exam preparation, and supported me in my TA role in EBS 165.

I am deeply grateful to my parents, Hangeum Son and Byounguk Yun, and my sister Jiin Yun for their endless love and support throughout my life. I thank my wife's parents, Myoungja Jo and Taemyoung Kim, for their warm encouragement and belief in me. Above all, I dedicate my boundless love and gratitude to my wife, Gayoung Kim. From the very beginning of my doctoral program to its conclusion today, she has been by my side, and she filled our time in Davis with happy and meaningful moments that I will cherish forever. Whenever I was exhausted and frustrated, she gave me strength with her words: "I'm sure you can do it."

Abstract

Vision models replace manual measurement of plants from an agricultural field, measuring plant traits with throughput beyond human capability and with greater consistency than manual methods. Recent vision models trained on web-scale datasets with billions of labeled examples have achieved near-human accuracy on natural image benchmarks. Yet, this scaling is unachievable for agricultural vision tasks where seasonal cropping cycles, sensor costs, and data annotation requirements restrict training data to orders of magnitude fewer samples. In breeding programs, for instance, evaluating thousands of genotypes across seasons forces a rigid trade-off between high-throughput, low-resolution remote sensing and low-throughput, high-resolution proximal measurement, typically yielding only thousands of images per season.

To resolve this bottleneck, this dissertation develops generative-model-based sensing methods that increase effective data resolution and throughput without proportionally increasing field labor or sensor costs: (1) thermal image enhancement for low-cost thermal cameras via image alignment and super-resolution; (2) plant architecture generation via vision language models; and (3) plot-level simulation generation via vision language models with in-context learning.

First, the resolution of low-cost thermal images was upscaled using pixel-level red–green–blue (RGB) and thermal alignment and a generative adversarial network (GAN)-based super-resolution algorithm. Thermal and RGB images were collected from a smartphone-mounted low-cost thermal camera. The collected RGB images were converted to pseudo-thermal images and aligned with low-resolution thermal images using template matching. Then the aligned RGB, pseudo-thermal, and low-resolution thermal images were fused and enhanced by the super-resolution algorithm, upscaling the resolution by a factor of four. Our results show a root mean squared error (RMSE) of 2.75 °C on temperature measure-

ment and a structural similarity index measure (SSIM) of 0.63 on the test set, and demonstrate recovery of leaf edge details from low-resolution thermal images by combining shape information from the RGB images.

Second, a plant architecture generation algorithm was developed using a vision language model with a vision encoder and a language decoder. Using a synthetic cowpea dataset generated from the Helios plant simulation library, we developed a tokenization method to serialize the nested structure of plant architecture and quantize continuous variables into discrete levels. Then, the vision language models were trained with 400,000 synthetic cowpea images and plant architecture pairs. The results show that our models can generate plant architecture from regular RGB images and outperform conventional regression-based methods in estimating leaf count and leaf area, with a mean absolute percentage error (MAPE) of 4.1% for leaf count and 3.2% for leaf area.

The final chapter expands the plant-level architecture generation to plot-level simulation generation using open-weight vision-language models with in-context learning. The plant simulation configuration includes metadata (year, location, plant type, days after planting), plant count, plant locations, solar position, and chlorophyll content, which are needed for Helios-based plant simulator to render the simulated plot. The open-weight models were evaluated on their ability to generate plant simulation configurations in JavaScript Object Notation (JSON) format from an image, using five different in-context learning methods. Five in-context learning methods were designed to provide information gradually, from basic format-restriction instructions, to few-shot examples, to grounding hints. Results show that the vision language models can generate plant simulation configurations from an image, and the Gemma 4 31B model with few-shot examples achieves an average symmetric mean absolute percentage error (sMAPE) of 19.4%.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	three-dimensional.
AI	artificial intelligence.
API	application programming interface.
B_{med}	median guess baseline.
B_{rnd}	random guess baseline.
BLEU-4	bilingual evaluation understudy 4-gram.
CD	Chamfer distance.
CLIP	contrastive language-image pre-training.
CNN	convolutional neural network.
CSV	comma-separated values.
DAP	days after planting.
F1	F1 score (harmonic mean of precision and recall).
FSPM	functional structural plant model.
GAN	generative adversarial network.
HTML	Hypertext Markup Language.
JSON	JavaScript Object Notation.
LiDAR	light detection and ranging.
LLM	large language model.
LoRA	low-rank adaptation.
LSTM	long short-term memory.
MAPE	mean absolute percentage error.
NCC	normalized cross correlation.
NeRF	neural radiance fields.
NETD	noise equivalent temperature difference.
NIR	near-infrared.
NTP	next token prediction.

PAR	photosynthetically active radiation.
PEFT	parameter-efficient fine-tuning.
PSNR	peak signal-to-noise ratio.
RANSAC	random sample consensus.
RGB	red–green–blue.
RMSE	root mean squared error.
SfM	structure from motion.
SIFT	scale-invariant feature transform.
sMAPE	symmetric mean absolute percentage error.
SR	super-resolution.
SSIM	structural similarity index measure.
UAV	unmanned aerial vehicle.
VFM	vision foundation model.
VisTA-SR	Visual & Thermal Alignment and Super-Resolution Enhancement.
ViT	vision transformer.
VLM	vision language model.
WD	Wasserstein distance.
XML	eXtensible Markup Language.

Chapter 1

Introduction

In plant breeding programs that evaluate thousands of genotypes across multiple growing seasons, researchers increasingly rely on deep learning to automate plant trait measurement and to scale up phenotyping throughput. Yet training these models requires large-scale, agricultural-domain-specific data that are scarce and expensive to collect. Field data acquisition is constrained by resolution requirements, economic constraints, labor availability, weather conditions, and crop-cycle limitations, and data annotations require specialized expertise from plant breeders or agronomists.

Agricultural data collection confronts a fundamental trade-off between high-throughput, low-resolution remote sensing and low-throughput, high-resolution proximal measurement (Smith et al., 2021). Aircraft-based remote sensing captures larger areas per flight, but higher-altitude flights compress plant-level variation into a few pixels, creating a direct trade-off between resolution and field of view (Messina and Modica, 2020; Klapp et al., 2021). Conversely, low-altitude flights provide high-resolution results but with reduced area coverage and increased cost (Klapp et al., 2021). Ground-based phenotyping platforms such as rovers can capture plot-level data with high resolution. However, rovers generally require long traverse times to cover a field and cannot measure all plots simultaneously (Bao et al., 2019; Li et al., 2014). Therefore, agricultural image collection for plant phenotyping must choose between extensive spatial coverage and fine detail, because both cannot be achieved simultaneously with current systems.

Since accurate plant trait measurements such as flower counts, fruit counts, plant height and canopy volume require high-resolution images with fine detail, scaling this high-

resolution sensing faces an economic bottleneck. Multispectral cameras provide vegetation indices at moderate expense, yet thermal and light detection and ranging (LiDAR) systems demand substantially greater investment for specialized optics, calibration, and data-processing pipelines (Klapp et al., 2021; Li et al., 2014). Beyond the economic bottleneck, the operational burden extends to labor availability and environmental constraints. Intensive phenotypic measurement by trained experts, destructive harvest, and repeated manual observation are all limited by workforce availability. Weather conditions and annual cropping cycles further narrow the data-collection window into discrete intervals, so even experiments that capture thousands of images per season accumulate training sets many orders of magnitude smaller than the datasets used to train large-scale computer vision models.

Low-cost sensors offer greater spatial coverage through deployment at scale than expensive, high-cost alternatives (Hufkens et al., 2019). However, such affordability often comes at the expense of data quality (Klapp et al., 2021). Recent advances in computational imaging and machine learning have begun to bridge this gap through aligning images and enhancing image resolutions. For example, super-resolution can increase the resolution of red–green–blue (RGB) images (Ledig et al., 2017), monocular depth estimation recovers three-dimensional (3D) scene depth without additional hardware (Yang et al., 2024), and neural radiance fields generate detailed 3D geometry from sparse views (Kerbl et al., 2023).

While deep learning-based image enhancement algorithms augment the visual information of images, vision foundation models (VFMs) offer an alternative approach to enhancing the information by using pretrained foundational knowledge. VFMs learn joint image-text embeddings that bridge visual features and natural language (Radford et al., 2021; Agrawal et al., 2022). VFMs are trained on web-scale datasets scraped automatically from online sources and annotated with crowdsourced labels that lack expert curation. Because these web-scale sources contain negligible agricultural content, deploying VFMs for agri-

cultural phenotyping confronts a data-scarcity barrier. Closing this gap will require scaling agricultural datasets close to the level of web-scale vision corpora, an expansion constrained by the same economic, labor, and environmental bottlenecks.

The main objective of this dissertation is to develop generative-model-based sensing methods that increase image quality and throughput for plant analysis without proportionally increasing field labor or sensor cost. This dissertation addresses the following research questions: (1) Can generative models effectively upscale low-resolution thermal images to enhance plant temperature analysis? (2) Can vision language models (VLMs) generate organ-level plant architecture token sequences from a single RGB image? (3) Can VLM pipelines for generating plant architecture models be scaled to simulate entire breeding fields?

The specific contributions of this dissertation are as follows:

- A calibration and validation protocol for low-cost thermal cameras to improve temperature measurement accuracy in agricultural applications, combined with an integrated image alignment and super-resolution deep learning framework that enhances the spatial resolution of low-cost thermal images by fusing aligned RGB and thermal inputs (Chapter 3).
- A specialized plant architecture tokenization method that converts eXtensible Markup Language (XML)-based plant models into token sequences, and a VLM trained to generate organ-level architecture tokens from single RGB images, validated on a synthetic cowpea dataset of 400,000 image-architecture pairs (Chapter 4).
- An image-to-simulation benchmark that scales single-plant architecture generation to plot-level JavaScript Object Notation (JSON) configurations, evaluating six open-weight VLMs under five in-context learning methods on synthetic and real datasets (Chapter 5).

These contributions are published in the following papers: Chapter 3 as Yun et al., 2024, Chapter 4 as Yun et al., 2026a, and Chapter 5 as Yun et al., 2026b.

Chapter 2

Literature Review

2.1 Introduction

Chapter 1 identified two technical barriers to scalable plant phenotyping using image analysis: the trade-off between spatial resolution and spatial coverage in sensor acquisition, and the lack of computational methods capable of inferring high-fidelity plant properties from constrained image inputs without commensurate increases in hardware cost or field labor. These barriers operate across three scales (pixel, plant, and plot) and motivate the dissertation's focus on generative modeling for image-based plant phenotyping. This chapter reviews the scientific foundations that enable such generative approaches, tracing the evolution of methods for image understanding, image enhancement, plant modeling, vision models, large language models, and structured output generation from images. The review is organized into three thematic parts that mirror the dissertation's technical pillars.

The first part (Section 2.2) examines thermal sensing and image enhancement. Section 2.2.1 surveys infrared thermography in agriculture from high-end industrial cameras to low-cost alternatives, establishing that organ-level thermal phenotyping remains inaccessible at scale. Subsections on image processing, convolutional neural networks, generative adversarial networks, and diffusion methods trace the evolution of single-image super-resolution from classical interpolation to deep generative models, and Subsections 2.2.6 and 2.2.7 review multi-observation approaches and guided thermal super-resolution, respectively, showing why the former is impractical in the field and why the latter lacks radiometric validation for agricultural scenes.

The second part (Section 2.3) covers plant architecture, three-dimensional reconstruction, and simulation. Section 2.3.1 establishes why three-dimensional plant structure affects photosynthetic performance, while Section 2.3.2 reviews methods for measuring plant architecture through active and passive sensing, from terrestrial LiDAR to recent implicit neural representations. Section 2.3.5 summarizes functional-structural plant models that integrate plant geometric structure with biophysical function, and Section 2.3.8 identifies a gap in pipelines that translate images directly into structured, machine-readable simulation configurations.

The third part (Section 2.4) reviews the advancement of generative artificial intelligence (AI) toward structured output generation. Section 2.4.1 and Section 2.4.2 review the independent trajectories of vision models and language models, from convolutional networks and recurrent models to Transformer models. Section 2.4.3 and Section 2.4.5 examine how VLMs and grammar-constrained decoding bridge visual understanding and machine-readable simulation configurations, culminating in a critical observation: while VLMs have been extensively benchmarked on agricultural classification and visual question answering tasks, no prior work has tested their capacity to generate structured 3D plant simulation configurations from images.

Finally, Section 2.5 synthesizes the identified gaps into three cumulative research gaps, bridging directly to Chapters 3 to 5.

2.2 Thermal Imaging and Super-Resolution

2.2.1 Thermal Imaging in Agricultural Research

Infrared thermography is used in agricultural research because it enables non-invasive, non-contact measurement of plant temperature (Ishimwe et al., 2014; Messina and Modica, 2020). Because canopy temperature is inversely correlated with stomatal conductance and transpiration rate, thermal imaging is widely employed as a surrogate for plant water

status (Möller et al., 2007). Zia et al. (2012); Mangus et al. (2016) presented canopy-level temperature monitoring, employing the crop water stress index (CWSI) to quantify water stress of plants. Although thermal imaging has shown potential for organ-level applications such as evaluating fruit maturity and detecting bruises (Ishimwe et al., 2014), reviews of infrared thermography applications remain largely weighted toward canopy-scale studies of water stress.

Gonzalez-Dugo et al. (2013) showed the utility of high-resolution unmanned aerial vehicle (UAV) thermal imagery by assessing water stress within a commercial orchard using a Thermoteknix MIRICLE 307 camera (640×480 resolution, $20,000), establishing an early benchmark for high fidelity thermal phenotyping. Yan et al. (2023) recently employed a Workswell ProSC TIR camera (640×512 resolution, $17,250) mounted on a UAV to estimate evaporation, transpiration, and evapotranspiration across a subalpine wetland.

Low-cost thermal cameras can provide an alternative for agricultural applications. For example, García-Tejero et al. (2018) compared a low-cost FLIR One camera (80×60 resolution, $400) with a high-end FLIR SC660 camera (640×480 resolution, $20,000) for assessing crop water status in almond trees, finding that the low-cost camera provided reliable estimates comparable to the high-end device. Iseki and Olaleye (2020) used a FLIR C2 camera (80×60 resolution, $500) to develop a new thermal indicator of leaf stomatal conductance in cowpea, and Parihar et al. (2021) utilized a FLIR E6 camera ($2,000) for irrigation scheduling of potted hibiscus plants. However, low-cost cameras have limited levels of phenotypic detail because of substantially lower spatial resolution than high-end thermal cameras. Low-resolution thermal cameras blur organ-level differences into coarse canopy averages, limiting the usage of biophysical models that depend on fine-grained temperature inputs.

Table 2.1 summarizes the cost and resolution spectrum of thermal cameras. While these studies showed promising results, the cost of such industrial-grade thermal cameras limits

access to most research programs and farming operations. The low-cost alternatives trade spatial resolution and, in some cases, radiometric calibration for affordability, creating the gap addressed in Chapter 3.

Table 2.1. Thermal cameras used in the agricultural literature. All low-cost cameras deliver substantially lower resolution than their high-end counterparts.

Study	Task	Camera	Resolution	Price (Approx.)
Gonzalez-Dugo et al. (2013)	Water stress	Thermoteknix MIRICLE 307	640×480	>$20k
Yan et al. (2023)	Evap./transp.	Workswell ProSc TIR	640×512	$17,250
García-Tejero et al. (2018)	Water status	FLIR SC660	640×480	>$15k
García-Tejero et al. (2018)	Water status	FLIR One	80×60	<$500
Parihar et al. (2021)	Irrigation sched.	FLIR E6	240×180	$2,000
Iseki and Olaleye (2020)	Stomatal conduct.	FLIR C2	80×60	<$500

To overcome this hardware limitation without replacing sensors, Super-resolution (SR) is a computational approach to recover fine-grained temperature maps from a solitary low-resolution observation. Because the forward degradation process (blurring, downsampling, and compression) is not uniquely invertible, super-resolution is ill-posed, and any practical solution must rely on priors learned from training data. The following subsections trace the evolution of SR from classical interpolation through classical resampling, convolutional neural network (CNN)-based, generative adversarial network (GAN)-based, and diffusion methods.

2.2.2 Image Upscaling Methods

Classical image upscaling methods use interpolation and filtering kernels. For example, bilinear interpolation (Smith, 1981) computes weighted averages of neighboring pixels, producing smooth results but blurring sharp edges and high-frequency texture. Lanczos resampling (Duchon, 1979) uses windowed sinc kernels to reduce aliasing during interpolation.

But like other linear filter-based methods, they cannot recover high-frequency content that was lost during downsampling, missing fine detail beyond what is present in the input.

2.2.3 Convolutional Neural Network-Based Methods

Dong et al. (2016) introduced the super-resolution convolutional neural network (SRCNN), demonstrating that a three-layer network could learn an end-to-end mapping from interpolated low-resolution inputs to high-resolution outputs more accurately than hand-designed filters. Subsequent architectures increased depth and introduced residual connections to ease training. Kim et al. (2016) proposed the very deep super-resolution network (VDSR), a 20-layer convolutional network that exploited residual learning and adjustable gradient clipping to achieve large gains. Lim et al. (2017) introduced the enhanced deep super-resolution network (EDSR), a wide residual network that removed batch normalization layers to preserve range flexibility, further improving reconstruction fidelity. Zhang et al. (2018) developed the residual dense network (RDN), which combined residual dense blocks with dense local connections to fully exploit hierarchical features from all convolutional layers, while Dai et al. (2019) proposed the second-order attention network (SAN), introducing a second-order channel attention module that adaptively rescales features using higher-order feature statistics for more discriminative representations. Collectively, these CNN-based methods established that deeper networks with more sophisticated feature extraction progressively improved peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) metrics.

2.2.4 Generative Adversarial Network-Based Methods

While CNN-based methods minimize pixel-wise error, they often produce overly smooth images lacking realistic high-frequency textures. Ledig et al. (2017) addressed this limitation with the super-resolution generative adversarial network (SRGAN), the first framework to infer photo-realistic natural images at $4\times$ upscaling factors. By using a perceptual loss comparing high-level feature difference, SRGAN generated perceptually compelling de-

tails with substantial gains in mean-opinion-score testing. Wang et al. (2018) enhanced SRGAN by introducing the enhanced super-resolution generative adversarial network (ESRGAN), which uses residual-in-residual dense blocks (RRDB), removes batch normalization, and adopts a relativistic average discriminator, yielding sharper and more natural textures than SRGAN. Wang et al. (2021) improved ESRGAN to Real-ESRGAN, which trained a blind super-resolution network with pure synthetic data produced via a high-order degradation process to robustly deal with real-world noise, blur, and compression artifacts.

2.2.5 Diffusion-Based Methods

More recently, diffusion probabilistic models such as SR3 (Saharia et al., 2021) have demonstrated strong perceptual performance on natural images by iteratively denoising a noisy image state conditioned on a low-resolution input. Diffusion models require large computational resources, large training data, and careful conditioning mechanisms (Moser et al., 2024). Although recent studies have begun exploring diffusion models for generic thermal super-resolution (Cortés-Mendez and Hayet, 2024; Choi and Kim, 2025), these preliminary works have not addressed the specific radiometric and cross-domain requirements of agricultural scenes. Therefore, we restrict the scope of this dissertation to GAN-based and hybrid adversarial CNN methods, which currently represent the most extensively validated approach for agricultural thermal super-resolution.

2.2.6 Multi-Observation Thermal Super-Resolution

A distinct family of methods attempts to circumvent the ill-posedness of single-image upsampling by using multiple observations of the same scene. Mandanici et al. (2019) presented a multi-image super-resolution method for terrestrial thermal imagery, registering multiple low-resolution thermal frames of a building scene to reconstruct a single high-resolution thermal image. Their results demonstrate that sub-pixel alignment of multiple observations can recover finer geometric detail than any single frame alone. Yet their out-

Table 2.2. Evolution of single-image upscaling methods from classical resampling to deep generative models.

Generation	Method	Author & Year	Contributions
Resampling-based	Bilinear	—	Classical linear interpolation between nearest neighbors
	Lanczos	—	Sinc-windowed kernel for signal reconstruction
CNN-based	SRCNN	Dong et al. (2016)	First end-to-end CNN for SR; 3 conv layers
	VDSR	Kim et al. (2016)	Very deep CNN (20 layers) predicting residual images with gradient clipping
	EDSR	Lim et al. (2017)	Wide residual blocks; removed batch normalization
	RDN	Zhang et al. (2018)	Residual dense blocks and dense skip connections
	SAN	Dai et al. (2019)	Second-order channel attention for feature rescaling
GAN-based	SRGAN	Ledig et al. (2017)	Perceptual loss for photo-realistic 4× SR
	ESRGAN	Wang et al. (2018)	Residual-in-residual dense blocks + relativistic discriminator
	Real-ESRGAN	Wang et al. (2021)	High-order degradation model for real-world blur/noise
Diffusion	SR3	Saharia et al. (2021)	Iterative denoising conditioned on LR input

door survey also revealed the fragility of multi-image reconstruction to temporal change: over the course of a 25-minute acquisition, cooling of the scene and movement of clouds introduced unreal temperature patterns that degraded the reconstructed output. These findings illustrate that multi-image methods demand strict temporal coherence, forcing the scene to remain sufficiently static between acquisitions.

Rivadeneira et al. (2022) proposed a deep-learning-based multi-image super-resolution architecture for thermal images. Because of the limited availability of real multi-image thermal datasets, researchers synthesized low-resolution images by downsampling, shifting, and adding noise to high-resolution thermal images. While this demonstrates that neural networks can exploit multi-frame information for thermal SR, the reliance on simulated data emphasizes a practical reality: acquiring genuine multiple aligned thermal frames of the same agricultural scene is difficult in field settings.

Although multi-observation thermal super-resolution methods recover more detail than single-observation methods, they require multiple observations of the static target. This limits its use in handheld or UAV-based field phenotyping, where crops move with the wind and illumination changes between overpasses.

2.2.7 Guided Single-Image Thermal Super-Resolution

Super-resolution of thermal images presents challenges distinct from natural-image SR. Thermal sensors capture long-wave infrared radiation (typically 7.5–14.0 μm) rather than visible light; as a result, thermal images lack the fine textures and high-frequency edge gradients that guide natural-image SR methods. Compared to visible images, the IR image consists of a single channel and its gradients may differ, which implies that texture-driven hallucination from purely learned priors can be misleading for biological structures whose boundaries are defined by temperature contrast rather than intensity gradients.

Several researchers presented guided super-resolution algorithms that augment a single

low-resolution thermal observation with a co-registered high-resolution image from another modality. In satellite remote sensing, pan-sharpening algorithms fuse a simultaneously acquired high-resolution panchromatic image with lower-resolution multispectral bands to synthesize a high-resolution multispectral output (Dadrass Javan et al., 2021). These methods assume strong spatial and spectral correlations between co-registered bands, a condition that is met by design on satellite platforms but is difficult to replicate in agricultural field settings where thermal and visible sensors have different spectral responses, viewpoints, and optical axes. Although Raimundo et al. (2021) have transferred pan-sharpening techniques to UAV thermal imaging, the underlying spectral–spatial injection models were developed for satellite multi-band sensors and do not directly address the radiometric fidelity demands of crop temperature mapping.

In the agricultural domain, Klapp et al. (2021) demonstrated an end-to-end pipeline that jointly stabilizes low-cost thermal camera readouts and performs CNN-based super-resolution, reporting temperature RMSE improvements of an order of magnitude after mosaicking. Rivadeneira et al. (2021) presented the thermal image super-resolution challenge, which reflects growing community interest; however, the challenge dataset covers generic indoor and outdoor scenes rather than agricultural benchmarks, indicating that crop-specific evaluation protocols remain undeveloped.

Multimodal super-resolution (MMSR) assumes that edge and texture information in a high-resolution guide image can constrain the ill-posed thermal upsampling problem. Chen et al. (2016) and Almasri and Debeir (2018) demonstrated that registration of RGB and thermal images enables cross-modal feature propagation. Gupta and Mitra (2022) extended this to unaligned inputs using cross-modal attention and guided filtering, and showed that even misaligned RGB images can improve thermal SR outcomes. However, the misalignment, field of view, lens distortion, pixel density, and sensor position differences between RGB and thermal cameras introduce artifacts such as ghosting and edge

halos. Gupta and Mitra (2020) introduced pyramid edge-maps and attention to address these issues, but noted that significant differences in spectral response remain challenging.

Table 2.3. Thermal and multimodal super-resolution methods reviewed in Section 2.2. The Dataset column indicates the dominant evaluation scene type (General = indoor/outdoor non-agricultural; Agriculture = crop or field scenes; Building = urban structural scenes; UAV = unmanned aerial vehicle imagery).

Method	Image Domain	Study	Key Innovation	Dataset
Multi-observation Methods				
Registration-based	Thermal	Mandanici et al. (2019)	Multi-image registration for building scene thermal survey	Building
Burst fusion	Thermal	Rivadeneira et al. (2022)	Attention-based burst fusion with synthetic multi-frame inputs	General
Guided Methods				
Color-guided filter	RGB+T	Chen et al. (2016)	Color-guided filtering for thermal SR	General
RGB-guided GAN	RGB+T	Almasri and Debeir (2018)	GAN-based RGB-guided thermal SR (TIGAN)	General
Pan-sharpening fusion	Pan+MS	Raimundo et al. (2021)	Pan–multispectral fusion for UAV thermal imagery	UAV
End-to-end CNN	Thermal	Klapp et al. (2021)	Joint radiometric stabilization + CNN SR for agriculture	Agriculture
Pyramid attention	RGB+T	Gupta and Mitra (2020)	Pyramidal edge-map attention for guided thermal SR	General
Unaligned attention	RGB+T	Gupta and Mitra (2022)	Self-attention for unaligned RGB–thermal SR	General

While the methods in Table 2.2 trace the technical evolution of super-resolution from classical resampling to deep generative models, nearly all were originally validated on natural RGB images. Table 2.3 reveals that only a handful of works have extended these techniques to thermal imagery, and among the eight thermal and multimodal methods surveyed, only Klapp et al. (2021) have validated their approach on agricultural scenes. Even then, organ-level radiometric accuracy was not explicitly evaluated. This asymmetry ex-

poses a gap: no existing framework jointly enhances spatial resolution and preserves temperature fidelity at the organ level under agricultural canopy geometry. Chapter 3 fills this gap by developing an integrated alignment and super-resolution network that preserves radiometric accuracy in agricultural thermal imagery.

While the preceding sections address the challenge of recovering pixel-level temperature from low-resolution sensors, the complementary challenge of inferring plant architecture from limited visual observations remains necessary for scalable phenotyping. Both problems fall under the umbrella of generative modeling for image-based phenotyping: inferring high-fidelity plant characteristics from constrained image inputs. The following sections review how plant structure has been measured, modeled, and simulated, tracing a progression from explicit 3D scanning to implicit generative representations that set the stage for vision-language-based inference.

2.3 Plant Architecture, Reconstruction, and Simulation

2.3.1 Plant Structure and Its Biological Consequences

The three-dimensional architecture of a plant governs how it captures light, exchanges gases, and converts solar energy into biomass. Consequently, canopy structure is a primary determinant of photosynthetic performance and crop yield. Experimental and simulation studies have repeatedly shown that structural traits derive direct functional consequences. Verbeeck et al. (2019) proposed that a structural economics spectrum, spanning crown density, branching angles, and crown dimensions, captures woody-plant functional diversity not represented by plant height alone. Niinemets (2010) summarized the effect of foliage inclination, spatial aggregation, and branching frequency on the canopy extinction coefficient, so the same leaf area index can support markedly different light-harvesting efficiencies depending on geometry. In maize, Duvick et al. (2003) documented a long-term trend toward more upright leaves over decades of breeding, which improves light distribu-

tion in dense canopies. Burgess et al. (2017) and Zhang et al. (2022) reinforced these links by showing that canopy structure is a primary driver of canopy light distribution and photosynthetic efficiency.

2.3.2 Three-Dimensional Plant Reconstruction

Reconstructing three-dimensional plant geometry is important in image-based phenotyping. The reconstruction methods can be categorized into two groups: explicit geometric reconstruction, which discretizes surfaces as point clouds, voxels, or meshes through active or passive sensing, and implicit neural representations, which learn continuous volumetric fields that jointly encode geometry and appearance (Li et al., 2025). The subsections below review both paradigms, tracing a progression from classical LiDAR and multi-view photogrammetry to recent neural approaches.

2.3.3 Explicit geometric reconstruction

Explicit geometric reconstruction methods for extracting 3D plant geometry fall into two broad categories: active and passive reconstruction methods. Li et al. (2025) summarized 3D reconstruction methods in plant phenotyping, reviewing active approaches such as LiDAR and structured light alongside passive techniques such as structure from motion (SfM) and photogrammetry. Terrestrial LiDAR has been used successfully to measure leaf area and leaf orientation at high resolution (Bailey and Mahaffee, 2017). SfM and photogrammetry reconstruct bulk plant geometry from overlapping images, enabling inexpensive data acquisition (Salter et al., 2021), while voxel carving offers a scalable alternative for reconstructing canopy architecture from small sets of RGB photos in controlled phenotyping environments (Gaillard et al., 2020).

Recent advances have pushed organ-level reconstruction further through graph-based skeletonization and instance segmentation from point clouds. Li et al. (2024b) proposed methods to segment individual stems and leaves from 3D point cloud data, and Ziamtsov and Navlakha (2020) introduced Plant3D, an open-source toolkit that performs classification,

skeletonization, and segmentation of plant point clouds to extract topological graphs and organ-level phenotypes. These developments have improved the granularity at which breeders can assign phenotypic traits to specific organs.

2.3.4 Implicit Neural Representations

More recently, implicit neural representations have emerged as alternatives to explicit geometric reconstruction. Neural radiance fields (NeRF) learn a continuous volumetric density and radiance field, enabling photo-realistic novel-view synthesis (Mildenhall et al., 2022). Based on this method, Saeed et al. (2023) used NeRF to reconstruct peanut plants, and Meyer et al. (2024) demonstrated FruitNeRF for fruit counting. Kerbl et al. (2023) introduced 3D Gaussian Splatting (3DGS), which represents scenes with anisotropic 3D Gaussians and achieves real-time rendering speeds. Plant-specific 3DGS applications include PlantGaussian for cross-time, cross-scene plant visualization (Shen et al., 2025), PlantDreamer for realistic plant generation via diffusion guidance (Hartley et al., 2025), and Splanting for 3D plant capture (Ojo et al., 2024).

2.3.5 Functional-Structural Plant Modeling

To overcome the physical limitations of measurement, plant modeling simulates biophysical processes such as photosynthesis and water use in silico. Functional-structural plant models (FSPMs) integrate a 3D geometric representation of plant architecture at the organ level with biophysical modeling of plant function (Sievänen et al., 2014). This integration enables the creation of plant models that reflect plant architectural development and physiological activity, facilitating the analysis of target phenotypes in breeding (Soualiou et al., 2021).

2.3.6 Formal Grammars and L-Systems

Architectural definitions for FSPMs usually derive from formal grammars. Lindenmayer (1968a) introduced L-systems, a parallel rewriting system that generates fractal-like

branching structures. Subsequent software tools embedded L-systems with organ-level parameters: L-Studio (Karwowski and Prusinkiewicz, 2004) and L-Peach (Allen et al., 2005) specify plant architectures using parameterized L-system modules, whereas XL (Kurth and Ong, 2017) generalizes these representations to a node-edge graph that encodes geometric and topological descriptors such as internode lengths, branching angles, and organ scales. Schnepf et al. (2018) developed CRootBox, an object-oriented root modeling system, and CPlantBox extended this to full plants (Zhou et al., 2020). Yun and Kim (2023) introduced CropBox, a declarative crop modeling framework that enables high-level specification of FSPMs using the Julia programming language.

2.3.7 Digital Twins and Modern Frameworks

Digital twins extend plant simulation by coupling a computational model to real-time measurement data, enabling AI-driven "what-if" experiments with various crops and environments (Ganapathysubramanian et al., 2025; Bellvert et al., 2023; Cesco et al., 2023; Escribà-Gelonch et al., 2024; Peladarinos et al., 2023). These digital twin models simulate growth under varying conditions, yet their parameters must be manually specified by domain experts from field observations or reconstructed geometry, limiting scalability for population-level studies.

2.3.8 Structured Configuration and the Automation Gap

Simulation pipelines require machine-readable structured formats, such as JSON or XML, to express plant definitions, environmental parameters, and input data. CPlantBox stores plant architectures in XML parameter files (Zhou et al., 2020). Lobet et al. (2015) proposed the Root System Markup Language, an XML-based standard for root architecture. Baker et al. (2024) used JSON commands to steer virtual plant scenes in Unreal Engine for synthetic data generation and visualization. Kim and Heo (2024) built an orchard-scale digital twin for mandarin production by aggregating multi-source environmental and crop data into structured records.

Despite powerful simulation capabilities, all current frameworks require manual specification of architectural parameters. No existing pipeline automatically infers complete organ-level architecture from a single image and encodes it into a structured, reproducible format suitable for simulation. Existing models primarily define individual plant architectures, and the extension to plot-level multi-plant configurations, including plant spacing, environmental context, and terrain, remains labor-intensive, largely manual, and unaddressed by any vision-language or generative model.

While Section 2.3.2 describes how 3D plant structure is reconstructed through sensors and algorithms, recent advances in artificial intelligence introduce a different approach: inferring structured plant representations directly from visual cues and natural language. Section 2.4.1 traces the trajectory of computer vision, which progressed through spatial feature hierarchies from convolutional nets to vision transformers. Section 2.4.2 reviews the parallel evolution of large language models, which developed independently through statistical language modeling. Although these two fields evolved along independent paths, both converged on the Transformer architecture as their foundational backbone, a convergence that Section 2.4.3 examines in detail by showing how vision encoders and language encoders can be aligned in a common representational space to yield vision-language models. Finally, Section 2.4.5 reviews structured document generation, the critical bridge that connects visual understanding to machine-readable simulation configurations. Together, these sections establish the foundation for the single-image plant architecture generation proposed in Chapter 4.

2.4 Vision-Language Models and Structured Output Generation

2.4.1 Vision Models

Computer vision models progressed from CNN-based models to Transformer-based models that learn general-purpose visual representations. Lecun et al. (1998) introduced LeNet-5,

a CNN that combined local receptive fields, shared weights, and spatial subsampling to extract visual features from images, outperforming hand-crafted features. These principles became foundational, but limited computational resources and dataset sizes constrained early convolutional networks to relatively compact architectures.

The modern deep learning era in computer vision began with Krizhevsky et al. (2012), who introduced AlexNet, a deep convolutional network of eight layers that won the ImageNet large-scale visual recognition challenge 2012 by a wide margin. AlexNet demonstrated that CNNs with sufficient training data could learn richly structured visual features that exceeded the performance of hand-crafted features or shallow classifiers. He et al. (2016) introduced ResNet, which addressed the degradation problem in very deep networks by learning residual mappings rather than direct mappings, enabling training of networks with hundreds or thousands of layers while maintaining optimization stability. Residual connections have since become a standard component in virtually all deep vision architectures.

More recently, the vision community has shifted toward Transformer-based architectures. Originally developed for natural language processing, the Transformer (Vaswani et al., 2017) replaced convolutional inductive biases with global self-attention. Vision transformer (ViT) (Dosovitskiy et al., 2021) demonstrated that a pure Transformer applied directly to image patches could achieve competitive performance when pre-trained on large datasets, eschewing the local connectivity assumptions of CNNs entirely. These Transformer-based vision models support foundation models such as DINOv2 (Oquab et al., 2024), which learns robust visual features through self-distillation, and the segment anything model (SAM) (Kirillov et al., 2023), which provides promptable segmentation for arbitrary objects. These vision models are capable of perceiving and segmenting the visual world without language. Yet pure vision alone does not interpret visual content through structured natural language, a limitation that would only be overcome once the two fields converged

on a shared representational backbone.

2.4.2 Language Models

Language modeling, which is the task of predicting the probability distribution over sequences of text, has advanced rapidly over the past decade. Early recurrent neural networks (RNNs) were limited by the vanishing and exploding gradient problems inherent to backpropagating through long sequential paths (Hochreiter and Schmidhuber, 1997). Long short-term memory (LSTM) networks were introduced to mitigate these problems, but the inherently sequential computation of recurrent models still precluded parallel training across long sequences and distant dependencies remained difficult to capture reliably.

Vaswani et al. (2017) introduced the Transformer architecture, which replaced RNNs with multi-head self-attention. The attention mechanism computes weighted interactions between all positions simultaneously, removing the sequential bottleneck of RNNs and enabling the capture of global dependencies. The original Transformer architecture used an encoder and a decoder, which map input sequences to continuous representations and generate output sequences conditioned on these representations. This architecture and its core components became the foundation of modern large language models (LLMs), most commonly in decoder-only autoregressive variants.

Scaling Transformer-based language models to billions of parameters and training on trillions of tokens produced LLMs with three defining characteristics. The large scale of the dataset enables LLMs to capture subtle statistical regularities in text. LLMs have foundational knowledge through unsupervised pre-training on web-scale corpora, allowing them to perform downstream tasks with minimal task-specific data. Also, LLMs can learn from context: the ability to adapt to new tasks within a single input prompt, without model weight updates. Brown et al. (2020) demonstrated that GPT-3, with 175 billion parameters, could achieve competitive few-shot performance on diverse natural language tasks by conditioning on only a handful of examples in the prompt.

Contemporary LLMs are released under two access strategies. Proprietary models such as the GPT series, Claude (Anthropic), and Gemini (Google DeepMind) are closed-weight and accessed only through commercial APIs, which relieves researchers from hardware ownership but introduces recurring costs, reliance on external service availability, and limited transparency regarding model architecture and training data. In contrast, open-weight models such as GPT-OSS (OpenAI), Gemma (Google), LLaMA (Meta) (Touvron et al., 2023), and Qwen (Alibaba) (Bai et al., 2023a) are distributed through public model hubs and can be run locally. The open-weight models can be fine-tuned, enabling domain-specific solutions for agricultural research with reproducibility and reduced inference costs. GPT-3 (Brown et al., 2020) established the scaling law with 175 billion parameters and text-only capability. Qwen-VL (Bai et al., 2023b) extends an open-weight backbone with vision capabilities.

2.4.3 Vision-Language Models

Language modeling (Section 2.4.2) and computer vision (Section 2.4.1) evolved along largely independent paths for decades: language models pursued statistical text prediction, while vision models pursued spatial feature hierarchies. Both trajectories converged, however, when the Transformer architecture became the shared building block. Because the same self-attention mechanism could process token sequences in text and patch sequences in images, separately trained language encoders and vision encoders could subsequently be projected into a common embedding space and aligned.

The central development in this convergence was contrastive language-image pre-training (CLIP) (Radford et al., 2021). CLIP demonstrated that models trained on 400 million image-caption pairs can acquire zero-shot classification, retrieval, and generalization capabilities by aligning visual and textual representations in a shared embedding space. Rather than requiring task-specific classifiers, CLIP learns a joint space where semantically similar images and texts cluster together, enabling open-vocabulary recognition without further

fine-tuning.

Vision-Language Models (VLMs) extend this alignment paradigm to generative tasks. BLIP (Li et al., 2022) introduced bootstrapping language-image pre-training that unified understanding and generation tasks. LLaVA (Liu et al., 2023c) introduced visual instruction tuning, enabling VLMs to follow natural-language instructions about image content by projecting CLIP visual features into the language model's input space. Qwen-VL (Bai et al., 2023b) added OCR, bounding-box, and multilingual capabilities, extending the breadth of visual reasoning tasks that can be addressed. Concurrently, proprietary frontier models including GPT-4o (OpenAI et al., 2024) and Gemini with vision have demonstrated strong performance on visual question answering, chart understanding, and spatial reasoning.

These models typically employ cross-modal attention layers that allow the language decoder to attend to spatially grounded visual tokens from an image encoder. This architecture enables VLMs to not only classify or caption images, but to count, localize, and reason about spatial relationships within them. However, the application of VLMs to agriculture has largely been limited to classification, disease identification, and visual question answering. None of the existing VLM architectures have been extended to generate structured, simulation-ready plant configurations from images, and no existing benchmark evaluates multi-task structured generation of 3D plot simulation configurations.

2.4.4 Agricultural VLM Benchmarks

In the agricultural domain, specialized benchmarks have emerged to evaluate the zero-shot and few-shot capabilities of VLMs. Zhou et al. (2024) suggested a vision-language framework for few-shot crop disease classification from Qwen-VL-generated prompts and attention-based feature enhancement within CLIP. Yang et al. (2025) trained AgriGPT-VL and evaluated it on AgriBench-VL-4K, which contains agricultural visual question answering and image-grounded single-choice questions. Awais et al. (2025) trained AgroGPT

for multi-turn multimodal agricultural dialogue and assessed it using the AgroEvals benchmark. Shinoda et al. (2025) introduced AgroBench, an expert-annotated benchmark spanning seven agricultural tasks: disease, pest, and weed identification, crop management, disease management, traditional methods, and machinery usage. Arshad et al. (2025) proposed the AgEval benchmark for plant stress phenotyping, covering twelve tasks across identification, classification, and quantification of stresses, diseases, pests, and seed quality. Wang et al. (2024a) introduced Agri-LLaVA, a knowledge-infused pest and disease VLM. Despite this proliferation, no existing benchmark evaluates multi-task structured generation of plant simulation configurations. A benchmark is needed that encompasses plant-type classification, environmental factor regression, plant localization, and biophysical parameter regression within a single generation task.

2.4.5 Structured Document Generation

The ability of language models to generate structured outputs such as JSON, XML, comma-separated values (CSV), and Hypertext Markup Language (HTML) bridges raw visual analysis and downstream simulation pipelines. Recent advances have enabled models to understand schemas and reproduce structured documents from visual or textual context (Dong et al., 2024; Geng et al., 2023). By combining these structured output capabilities with visual understanding, VLMs can generate machine-readable representations from images, ranging from JSON configuration files to 3D object descriptions.

Vision models can count and localize objects within images (Kang et al., 2024; Subramanian et al., 2022). However, unconstrained autoregressive generation is prone to generating false structured documents that have broken document structures or wrong syntax. To mitigate this, structured decoding limits the vocabulary at each step using a finite-state automaton or context-free grammar that encodes the target syntax (Dong et al., 2024; Geng et al., 2023). When combined with visual understanding, these constrained generation techniques enable VLMs to convert images into structured, machine-readable representations.

Kim et al. (2022) converted document images into structured text outputs such as JSON without optical character recognition. Lee et al. (2022) proposed Pix2Struct for parsing webpage screenshots into simplified HTML as a pretraining objective. Liu et al. (2023a) introduced DePlot for converting chart images into text tables, and Liu et al. (2023b) introduced MatCha, which enhances visual language pretraining with chart derendering and math reasoning tasks. Wang et al. (2024b) demonstrated that an LLM can generate 3D objects in plain-text mesh definitions from text prompts. These findings that language models can generate structured documents rather than natural language alone motivate the plant architecture generation approach proposed in Chapters 4 and 5.

2.5 Synthesis and Research Gaps

This literature review covered thermal imaging, image super-resolution, 3D plant phenotyping, plant modeling, large language models, and vision-language models. Across all three nested scales of image-based plant analysis, existing methods provide partial solutions that do not meet the scalability constraints of plant analysis and phenotyping. These limitations are organized into the following three cumulative research gaps, which motivate the studies presented in Chapters 3 to 5.

Research Gap 1 (Pixel-level): No validated method exists to improve the resolution and radiometric accuracy of low-cost thermal cameras for organ-level crop phenotyping. Agricultural thermal super-resolution is an ill-posed inverse problem in which thermal images lack the texture priors that guide natural-image SR methods. Existing multimodal methods align and fuse visible and thermal data, but have not been tested on agricultural canopy scenes with complex geometry, variable lighting, and large resolution disparities. Chapter 3 addresses this gap by developing Visual & Thermal Alignment and Super-Resolution Enhancement (VisTA-SR), an integrated image alignment and super-resolution framework that uses paired RGB and thermal images to enhance the spatial resolution and radiometric accuracy of low-cost thermal cameras, validated specifically on agricultural

organ-level tasks.

Research Gap 2 (Plant-level): No existing pipeline automatically translates a single agricultural image into a structured, reproducible plant simulation configuration. Existing 3D reconstruction methods require specialized equipment or multi-view image sets that are impractical at scale. While recent vision foundation models and VLMs have advanced image understanding, no study has used a VLM to generate token sequences that define a plant's architectural topology, geometry, and biophysical parameters. Chapter 4 addresses this gap by training a vision-language model to predict plant architecture tokens from synthetic images, serving as a proof of concept for single-image plant configuration generation.

Research Gap 3 (Plot-level): No existing framework scales single-plant architecture generation to plot-level simulation configurations for digital twin and breeding applications. Individual plant modeling tools such as CPlantBox and CropBox enable detailed single-plant simulation, but translating these to field scales involves plant spacing, terrain, environmental context, and management parameters that are still specified manually. Chapter 5 addresses this gap by exploring the scalability of image-to-plant-architecture methods toward plot-level multi-plant simulation configurations.

Chapter 3

VisTA-SR: Improving the Accuracy and Resolution of Low-Cost Thermal Imaging Cameras for Agriculture

Abstract

Thermal cameras are an important tool for agricultural research because they allow for non-invasive measurement of plant temperature, which relates to important photochemical, hydraulic, and agronomic traits. Utilizing low-cost thermal cameras can lower the barrier to introducing thermal imaging in agricultural research and production. This paper presents an approach to improve the temperature accuracy and image quality of low-cost thermal imaging cameras for agricultural applications. Leveraging advancements in computer vision techniques, particularly deep learning networks, we propose a method, called Visual & Thermal Alignment and Super-Resolution Enhancement (VisTA-SR) that combines RGB and thermal images to enhance the capabilities of low-resolution thermal cameras. The research includes calibration and validation of temperature measurements, acquisition of paired image datasets, and the development of a deep learning network tailored for agricultural thermal imaging. Our study addresses the challenges of image enhancement in the agricultural domain and explores the potential of low-cost thermal cameras to replace high-resolution industrial cameras. Experimental results demonstrate the effectiveness of our approach in enhancing temperature accuracy and image sharpness, making thermal imaging more accessible for agricultural research.

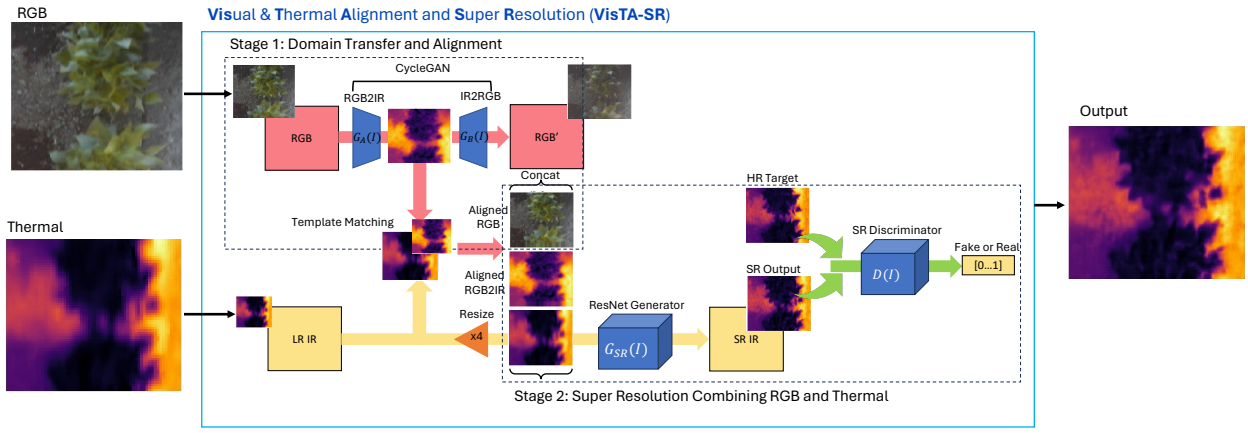


Figure 3.1. Structure of the proposed VisTA-SR network. The network has two main stages: the Image Alignment and the Super-Resolution Network. The Image Alignment aligns the RGB and thermal images, while the Super-Resolution Network enhances the resolution of the thermal image.

3.1 Introduction

Agricultural research often uses crop temperature measurement to detect abnormal plant characteristics, calculate crop water stress indices, or model complex biophysical interactions. Since various methods have been attempted to measure crop temperature, thermal imaging cameras are widely used because they can quickly measure the temperature at many points in the image (Khorsandi et al., 2018). Also, thermal imaging can quickly and non-invasively measure crop temperature compared to other temperature measurement devices. These benefits can help identify areas where crops are experiencing disease or stress, allowing for timely intervention.

Previous studies using thermal cameras in agriculture have utilized high-resolution industrial-grade thermal cameras (Gonzalez-Dugo et al., 2013). However, these cameras are very expensive, often costing over $10,000, which limits their accessibility. This low accessibility can restrict the widespread deployment of thermal cameras in agriculture, especially for researchers who cannot afford costly sensors.

An alternative approach is to use low-cost sensors. Recent developments in thermal image sensors and image processing technologies have made various affordable consumer-

grade thermal cameras available. These thermal cameras have the advantages of being relatively lightweight and easy to operate. Therefore, there have been attempts to use low-cost thermal cameras in agriculture (Giménez-Gallego et al., 2021, 2023; Blaya-Ros et al., 2020). For example, Bhandari (2016) obtained an image mask from visible light images and applied it to thermal images to measure wheat canopy temperature and estimate water stress. Another study used a low-cost thermal camera to calculate crop canopy temperature automatically (Giménez-Gallego et al., 2023). However, these low-cost thermal cameras have not been able to completely replace high-resolution industrial thermal cameras due to their lower pixel count and resolution.

Thermal camera resolution has a significant impact on the capability and accuracy of agricultural research. For example, low-resolution thermal cameras may only be able to recognize crops at the plant-level rather than organ-level, making it challenging to observe temperature differences between leaves, stems, flowers, and fruits, for instance. This low feature resolution limits the temperature measurement capability at various phenological stages, which is essential for developing precise crop biophysical models. Therefore, improving the quality of low-cost thermal images can increase the feasibility of using low-cost thermal cameras in agriculture.

Enhancing the resolution of low-resolution thermal images is a challenging task. It is an ill-posed problem, as multiple high-resolution ground truths can exist for a single low-resolution image. Nevertheless, various computer vision and machine learning techniques have been proposed to overcome the challenge. Particularly with the recent advancements in deep learning, there have been many reported cases of upsampling low-resolution images to high-resolution. Some researchers have used ResNet and GANs to perform image super-resolution (Ledig et al., 2017). Others have combined multiple low-resolution images to create a single high-resolution image (Rivadeneira et al., 2022). Some have also used multimodal data to improve the resolution of the data (Gupta and Mitra, 2022; Almasri and

Debeir, 2018; Chen et al., 2016). However, research on improving the quality of thermal images in the agricultural domain has been limited. Applying these techniques to agricultural thermal images may improve the image quality of low-resolution thermal cameras, allowing them to replace high-resolution thermal cameras.

Therefore, this paper studies how computer vision techniques can improve the image quality of low-resolution thermal cameras for agricultural applications. We propose a deep learning network that leverages complementary information from RGB and thermal image domains for both image alignment and super-resolution enhancement.

The specific contributions of this paper are as follows:

- Calibration and validation of the temperature measurement of a low-cost thermal camera in the agricultural domain
- Acquisition of a paired low-resolution thermal camera image dataset, as well as RGB and high-resolution thermal camera data in the agricultural domain
- Proposal of an integrated image alignment and super-resolution deep learning algorithm to improve the image quality of low-resolution thermal cameras by combining RGB and thermal images

3.2 Materials and Methods

3.2.1 Thermal Cameras

In this study, three types of thermal cameras were utilized. Table 3.1 shows the specifications of the thermal cameras. The VarioCAM HD camera, known for its high spatial resolution and temperature accuracy, was primarily used to create a dataset for temperature accuracy validation. The VarioCAM HD images were collected using their proprietary software on the Windows Operating System. The FLIR Boson camera, with a resolution of

Table 3.1. Specifications of thermal cameras used in this study

Specification	VarioCam HD Head 800	FLIR Boson	FLIR One Pro
Spectral Range	7.5 – 14 µm	8 – 14 µm	8 – 14 µm
Detector Resolution	1,024 × 768	640 × 512	160 × 120
Temperature Range	−40 – 2,000 °C	Non-Radiometric	−20 – 120 °C
Measurement Accuracy	±1.5°C or ±1.5%	Non-Radiometric	±3°C or ±5%
Sensitivity (NETD)	30 mK	40 mK	70 mK
Frame Rate	30 Hz & 60 Hz	9 Hz	8.7 Hz
Dimensions (mm)	221 × 90 × 94	21 × 21 × 11	68 × 34 × 14
Weight	1.15 kg	21 g	36.5 g
Price (Approx.)	$20,000	$4,000	$400

640×512, was employed to capture high-resolution thermal image data in the field. Positioned between high-end and consumer-grade thermal cameras in terms of price, the FLIR Boson camera offered a lightweight form factor and flexible video output interface for easy field image capture. FLIR Boson images were collected from the ROS-based system on Ubuntu PC. Lastly, the FLIR One Pro, a low-cost and low-resolution thermal camera, was used in this study. It has a thermal resolution of 160×120 and an RGB camera resolution of 1440×1080. FLIR One Pro image acquisition and storage were performed using a custom Swift-based app developed with the FLIR Mobile application programming interface (API) on an iPhone.

3.2.2 Low Cost Thermal Camera Calibration

Temperature measurement accuracy is important in agricultural applications of thermal imaging cameras. The sensor sensitivity of thermal imaging cameras is measured by the noise equivalent temperature difference (NETD), and even low-resolution thermal imaging cameras have NETD values of around 150 mK, which is sufficient for most agricultural applications. However, based on their specifications, there can be absolute errors of up to 3 °C in the images captured by low-resolution thermal imaging cameras, so absolute temperature accuracy must be verified.

Radiometric thermal cameras have a logarithmic relationship between the digital number and temperature (Tattersall, 2016; Minkina and Dudzik, 2009). The parameters for converting the digital number to temperature are stored in the EXIF tag information of the FLIR radiometric JPEG images. These parameters, which are pre-calibrated values from the factory, are used to convert the digital numbers of the thermal imaging camera to temperatures using Eq. (3.1). Upon comparing the factory parameters of different thermal imaging cameras, it was observed that only the values of R_1 and O differed, while the values of R_2, B, and F remained constant. The parameter B is derived from the Planck constant h and Boltzmann constant k_b, and the parameter F value is 1. For the FLIR One Pro cameras, R_2 was fixed at 0.0125, and R_1 and O are empirically calibrated depending on the individual camera.

$$\text{Temperature } (^{\circ}\text{C}) = \frac{B}{\ln(\frac{R_1}{R_2(DN+O)}) + F} - 273.15 \tag{3.1}$$

However, the accuracy of these factory parameters cannot be fully trusted as the manufacturer does not fully guarantee the temperature accuracy of the low-cost thermal cameras. To ensure the accuracy of temperature measurements, it is necessary to recalibrate the parameters of the thermal imaging camera. Therefore, the optimization process focused on optimizing the values of R_1 and O. The optimization was performed using the Nelder-Mead method (Nelder and Mead, 1965), which is a widely used optimization algorithm, with a tolerance of $1e-6$. The optimization process was implemented using the "scipy.optimize.minimize" function in Python.

Experiments were conducted to verify the temperature accuracy of the FLIR One Pro thermal imaging camera. The surface temperature of a controlled water bath was measured using a thermocouple with a digital data logger. The thermocouple measured the temperature starting from 4.0 °C, the initial temperature of the cold water, and reaching 100.0 °C, the boiling point of water. The air temperature and relative humidity were main-

tained during the experiment at 24.0 °C and 40%.

Fig. 3.2 shows the temperatures calculated using the factory parameters. The results showed that the temperatures calculated using the factory parameters were higher than reference temperatures below 30 °C. However, at temperatures near the boiling point of water, the measured temperatures were almost 20 °C lower than the actual temperatures. This indicates that the low-cost thermal camera's temperature values are inaccurate, especially at high temperatures.

The original and optimized parameters are shown in Table 3.2, and the temperatures calculated using the optimized new parameters are shown in Fig. 3.2. The temperatures calculated using the new parameters are more accurate than the results using the factory parameters, and they are almost identical to the temperatures measured by the thermocouple (Fig. 3.3).

Table 3.2. Comparison of factory and lab calibrated parameters

	R_1	B	F	O	R_2
Factory	18333.4	1435	1	-2284	0.0125
Optimized	12755.4	1435	1	-6707	0.0125

3.2.3 Matching Low-Resolution and High-Resolution Thermal Imaging Cameras

As part of a more extensive set of breeding experiments, Cowpea (*Vigna unguiculata L. Walp.*) and Common Bean (*Phaseolus vulgaris*) images were collected from June to September 2022 in Davis, California, to obtain high-resolution and low-resolution thermal images in the field. To match the low-resolution and high-resolution thermal imaging datasets, camera calibration was performed to calculate each camera's intrinsic parameters, and camera extrinsics were also measured. The two cameras were installed at a height of approximately 1.5 m from the ground, and the distance between the centers of the two cam-

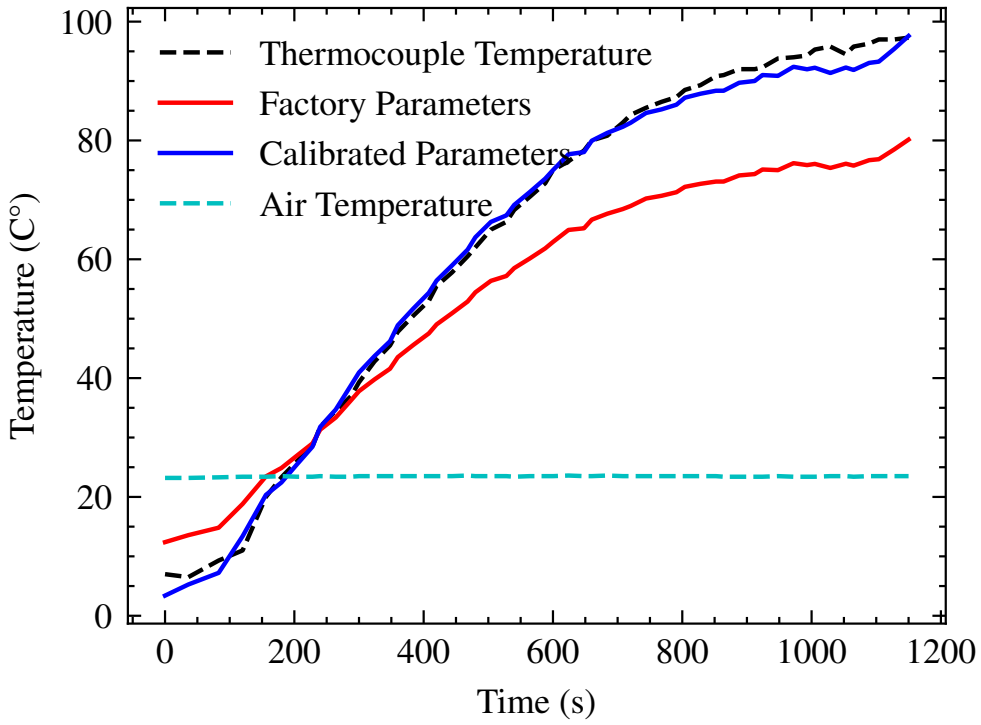


Figure 3.2. Comparison between thermocouple, factory, and calibrated temperature values in a time series

era lenses was 5cm.

However, the high-resolution and low-resolution thermal images were captured on different platforms at different frame rates, so matching the two datasets was challenging. Initially, scale-invariant feature transform (scale-invariant feature transform (SIFT)) (Lowe, 2004) feature extraction and matching were attempted, similar to the previous temperature accuracy test. However, the quality and the number of the extracted features in the images sometimes incorrectly estimated the homography between the low-resolution and high-resolution pair, which led to unstable matching and alignment results.

Since the field of view difference between the two images is only due to the scale difference based on the image resolution and the transitional offsets caused by the capture timings, template matching (Sarvaiya et al., 2009) was performed to robustly match the images by setting the high-resolution image as the template image T and calculating the normalized

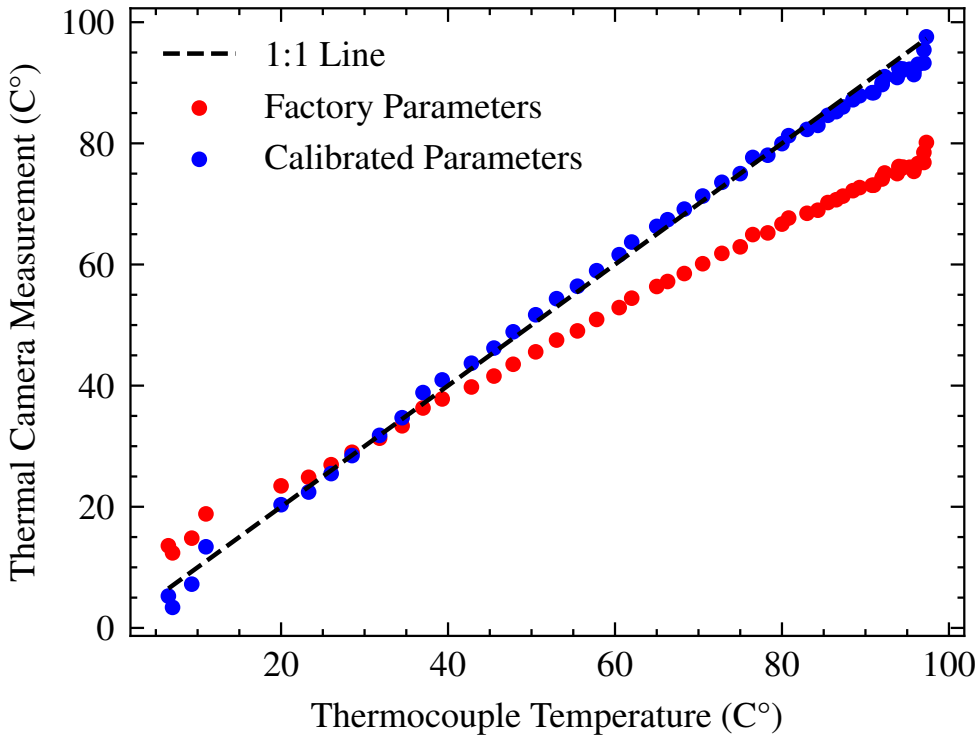


Figure 3.3. Comparison between factory and calibrated temperature values in a 1:1 plot

cross correlation (NCC) (Briechle and Hanebeck, 2001) between the template image and the low-resolution image I, finding the coordinates x^* and y^* where the NCC value was maximized, and resizing the template image to a predefined scale for this process.

$$R(x,y) = \sqrt{\frac{\sum_{x',y'}(T(x',y') - I(x+x',y+y'))^2}{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{3.2}$$

$$(x^*, y^*) = \underset{\substack{0 \le x < M \\ 0 \le y < N}}{\operatorname{argmax}} R(x,y) \tag{3.3}$$

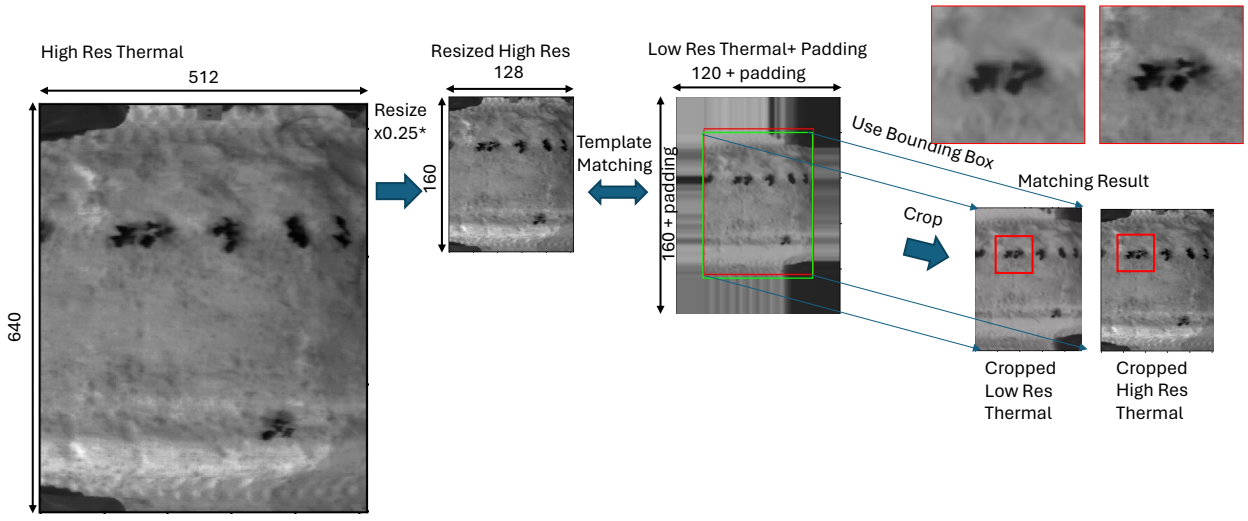


Figure 3.4. Matching and aligning process of low-resolution and high-resolution thermal images

The template matching was performed using Python OpenCV code. Fig. 3.4 illustrates matching the low-resolution and high-resolution thermal imaging cameras. For cases where the NCC value was 0.75 or higher, the bounding box was calculated and limited to the area within the padding of the low-resolution thermal image coordinate system. Then, it was converted to the coordinate system before resizing the template image. Image cropping was performed using the original resolution of the template and background images. The FLIR One Pro also has an integrated RGB camera, allowing simultaneous acquisition of RGB images. Therefore, the RGB images were also cropped using the template matching results.

3.2.4 Improving Image Resolution by Combining RGB and Thermal Imaging

In this paper, complementary information from the RGB image's structural details and the thermal image's intensity information is utilized to enhance the resolution of the low-resolution thermal camera. The RGB and thermal images obtained from the FLIR One Pro have the same field of view, but they are not perfectly pixel-aligned due to differences in camera lens position and video stream delays, which poses a challenge in combining the two modalities for resolution improvement. We tested deep-learning based image reg-

istration methods such as Spatial Transformer Networks (Jaderberg et al., 2015) and Deformable Field-based approaches (Zou et al., 2022). However, those methods tended to learn a shortcut existing in the dataset, which is a mean offset of the images rather than the differences between the input images, resulting in unstable experimental results.

Therefore, a template matching method based on image intensity was employed to align the domain-transformed image and the thermal image, which yielded more stable results compared to other methods. Fig. 3.5 illustrates the input RGB image, the RGB-to-thermal image translated by CycleGAN, and the low-resolution thermal image to be aligned. Inspired by the approach of Arar et al. (2020), the RGB image was first translated to the thermal imaging camera's domain using CycleGAN (Zhu et al., 2017). Then, template matching was performed between the domain-translated RGB image and the input low-resolution thermal image. The maximum correlation value was calculated based on the image convolution operation from one image to another, which can be hardware-accelerated and integrated into a super-resolution module using PyTorch.

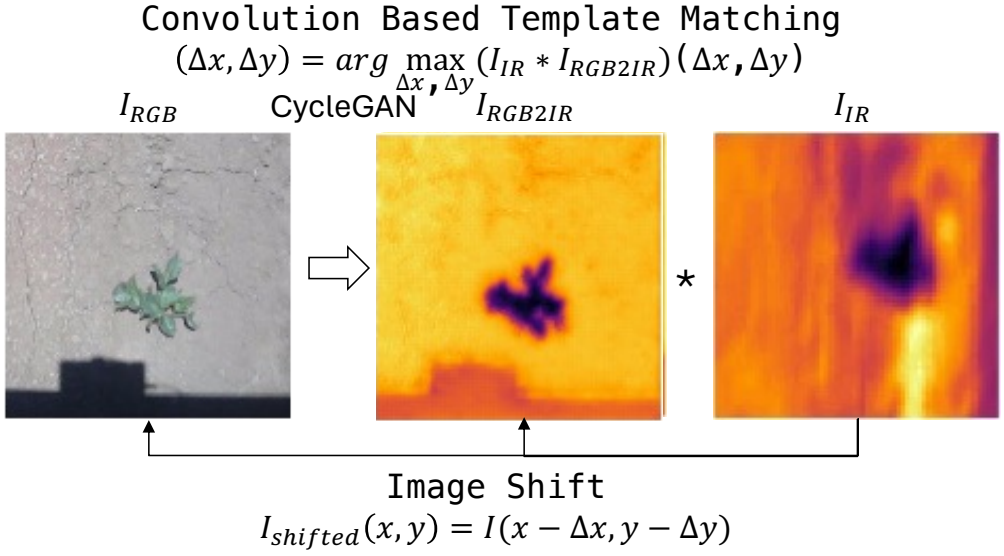


Figure 3.5. Matching RGB and thermal images using CycleGAN and template matching

After aligning the domain-transformed RGB image with the thermal image, the original

RGB image was also transformed using the alignment result. Subsequently, the RGB, domain-transformed, and low-resolution thermal images were combined and fed into a ResNet-based CNN. The output image is then fed into a Discriminator CNN for GAN training. This architecture is depicted in Fig. 3.1, referred to as VisTA-SR.

Except for CycleGAN (Zhu et al., 2017) and Template Matching, the implementation followed that of SRGAN (Ledig et al., 2017) and ESRGAN (Wang et al., 2018), and the loss function used is as follows:

Cycle Consistency Loss (Zhu et al., 2017):

$$l_{\text{Consi}}^{\text{Cycle}} = |I_{\text{RGB}} - G_{\text{IR2RGB}}(G_{\text{RGB2IR}}(I_{\text{RGB}}))| \tag{3.4}$$

Identity Loss (Zhu et al., 2017):

$$l_{\text{MSE}}^{\text{Cycle}} = ||I_{\text{HR}} - G_{\text{RGB2IR}}(I_{\text{RGB}})|| \tag{3.5}$$

MSE (Mean Squared Error) Loss (Ledig et al., 2017; Wang et al., 2018):

$$l_{\text{MSE}}^{\text{SR}} = ||I_{\text{HR}} - G_{\text{SR}}(I_{\text{LR}}, I_{\text{RGB}})|| \tag{3.6}$$

Content Loss (Ledig et al., 2017; Wang et al., 2018):

$$l_{\text{VGG}}^{\text{SR}} = ||\phi_{\text{VGG}}(I_{\text{HR}}) - \phi_{\text{VGG}}(G_{\text{SR}}(I_{\text{LR}}, I_{\text{RGB}}))|| \tag{3.7}$$

Adversarial Loss (Ledig et al., 2017; Wang et al., 2018):

$$l_{\text{Adv}}^{\text{SR}} = -\log D_{\text{SR}}(G_{\text{SR}}(I_{\text{LR}}, I_{\text{RGB}})) \tag{3.8}$$

Total Loss:

$$l_{\text{total},G} = (l_{\text{Re}}^{\text{Cycle}} + l_{\text{MSE}}^{\text{Cycle}}) + (l_{\text{MSE}}^{\text{SR}} + l_{\text{VGG}}^{\text{SR}} + \alpha l_{\text{Adv}}^{\text{SR}}) \tag{3.9}$$

3.3 Results

3.3.1 Low-Cost Thermal Camera Field Validation with High Fidelity Thermocouple Camera

Field data was collected to validate the temperature accuracy of the low-resolution thermal imaging camera in a real-world environment with crops and soil. The data was collected in the Garbanzo bean (*Cicer arietinum*) field located in Davis, California. The ground truth temperature values were measured using a VarioCAM HD camera and compared with the temperature measured by the FLIR One Pro thermal camera, and a total of 170 image pairs were collected on April 5, 2022. Image feature points were extracted from both images using the SIFT (Lowe, 2004) feature extractor, and they were matched using the Flann matching algorithm (Marius and David G, 2009). Then, the homography between the two images was calculated, and outliers were removed using the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981). As a result, the temperature values from the corresponding points in the two images were compared (Fig. 3.6).

The matching result for the 170 image pairs is shown in Fig. 3.7, and Table 3.3 summarizes the results. It indicates that the temperature measurement accuracy was improved from $R^2 = 0.86$ to $R^2 = 0.89$ after calibration, and the root mean squared error (RMSE) was also improved from 1.52 °C to 1.40 °C. Since using the factory parameters tends to overestimate the temperature when it is below 20 °C, as shown in Fig. 3.3, the temperature values obtained using the factory parameters in Fig. 3.6 also showed higher temperature measurements than the actual temperatures.

Table 3.3 also indicates that when calculating RMSE and R^2 using only data between 15 °C and 30 °C, the temperature measurements with calibrated parameters showed bet-

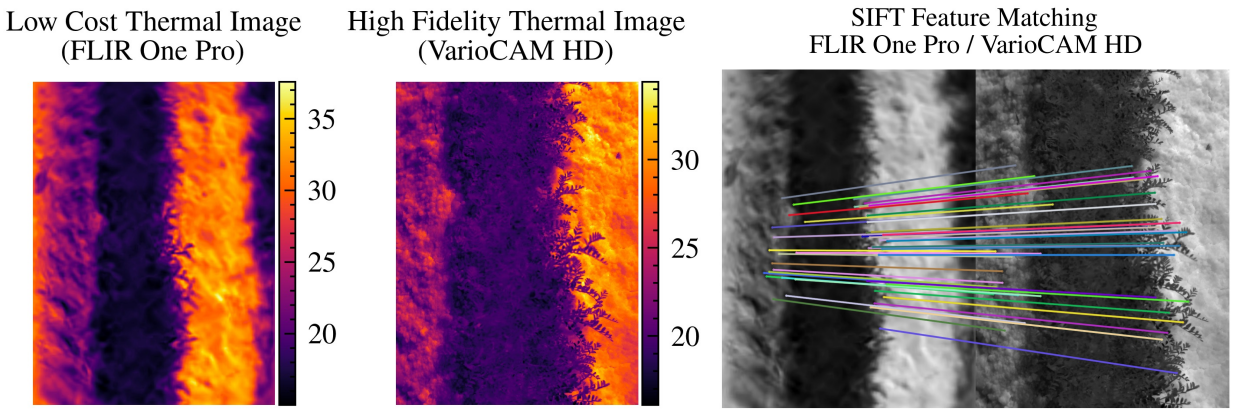


Figure 3.6. An example of feature matching based temperature comparison between FLIR One Pro and VarioCam HD Camera

ter accuracy. Considering the typical leaf temperature of plants, the accuracy within this temperature range is crucial for thermal cameras used in agriculture. Therefore, the thermal camera calibration in this study demonstrates the potential to enhance temperature measurement accuracy in agricultural research.

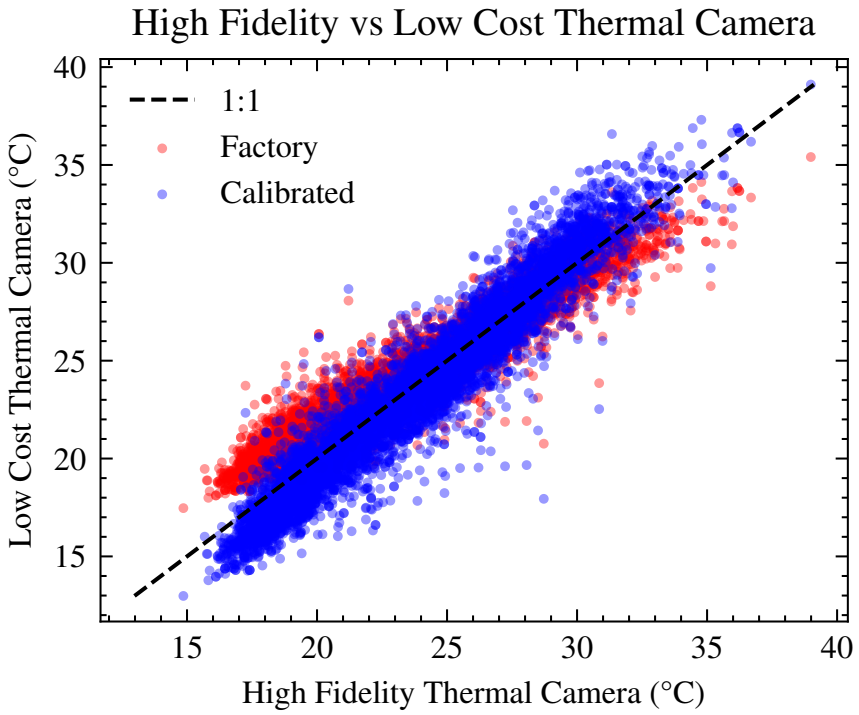


Figure 3.7. Feature matching based temperature comparison result. Plotted all matched temperature points for a total of 170 images

Table 3.3. Low-cost thermal camera (FLIR One Pro) temperature accuracy validation result before and after parameter calibration

	All data		**15 °C – 30 °C**	
	R^2	RMSE (°C)	R^2	RMSE (°C)
Factory	0.86	1.52	0.83	1.52
Calibrated	0.89	1.40	0.86	1.39

3.3.2 VisTA-SR Result

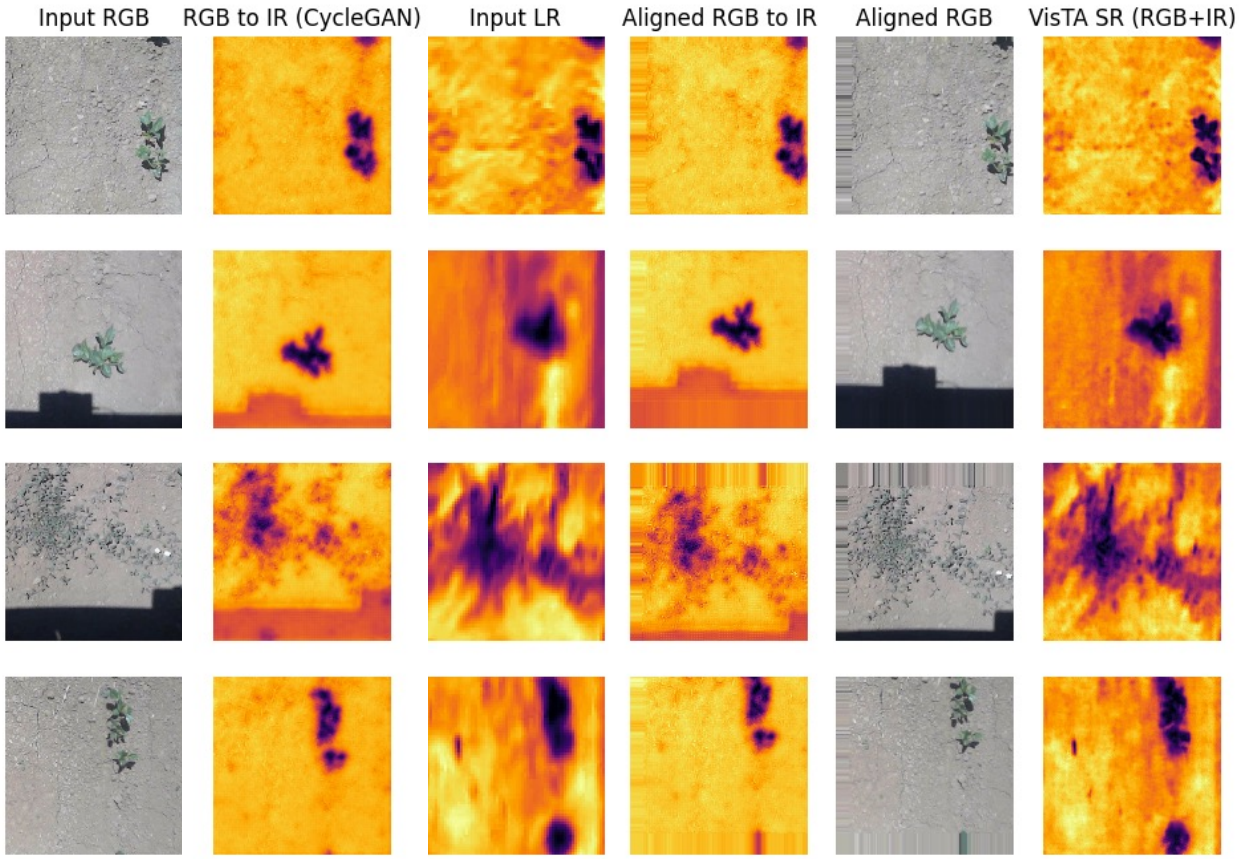


Figure 3.8. Input low-resolution images, domain-translated images, and aligned images produced by CycleGAN and template matching

In 2022, a total of 2612 image pairs were collected from a warm-season grain legume field across the growing season by matching low-resolution (160×120, FLIR One Pro) thermal images with high-resolution (640×512, FLIR Boson) thermal images. 80% of these pairs were used as training data, while the remaining 20% were used for validation. The network

was trained over 200 epochs with a batch size of 4. Fig. 3.8 demonstrates the image conversion quality and image alignment performance of the CycleGAN module, which was trained simultaneously with the SR Network. As depicted in the example images, CycleGAN successfully translated the image domain and template matching successfully aligned low-resolution thermal images based on image intensity.

Fig. 3.9 presents the results from the multiple input image scales obtained from the VisTA-SR algorithm using the input of the combined RGB image aligned with CycleGAN and Template Matching, compared to the results of the Super-Resolution Generative Adversarial Network (SRGAN) algorithm (Ledig et al., 2017; Wang et al., 2018) that utilizes only the existing thermal image modality. Our VisTA-SR demonstrated higher sharpness by leveraging higher-frequency structural information from the RGB image. This demonstrates that VisTA-SR improved the performance of capturing thermal properties of smaller features at the organ level, as opposed to the plant level.

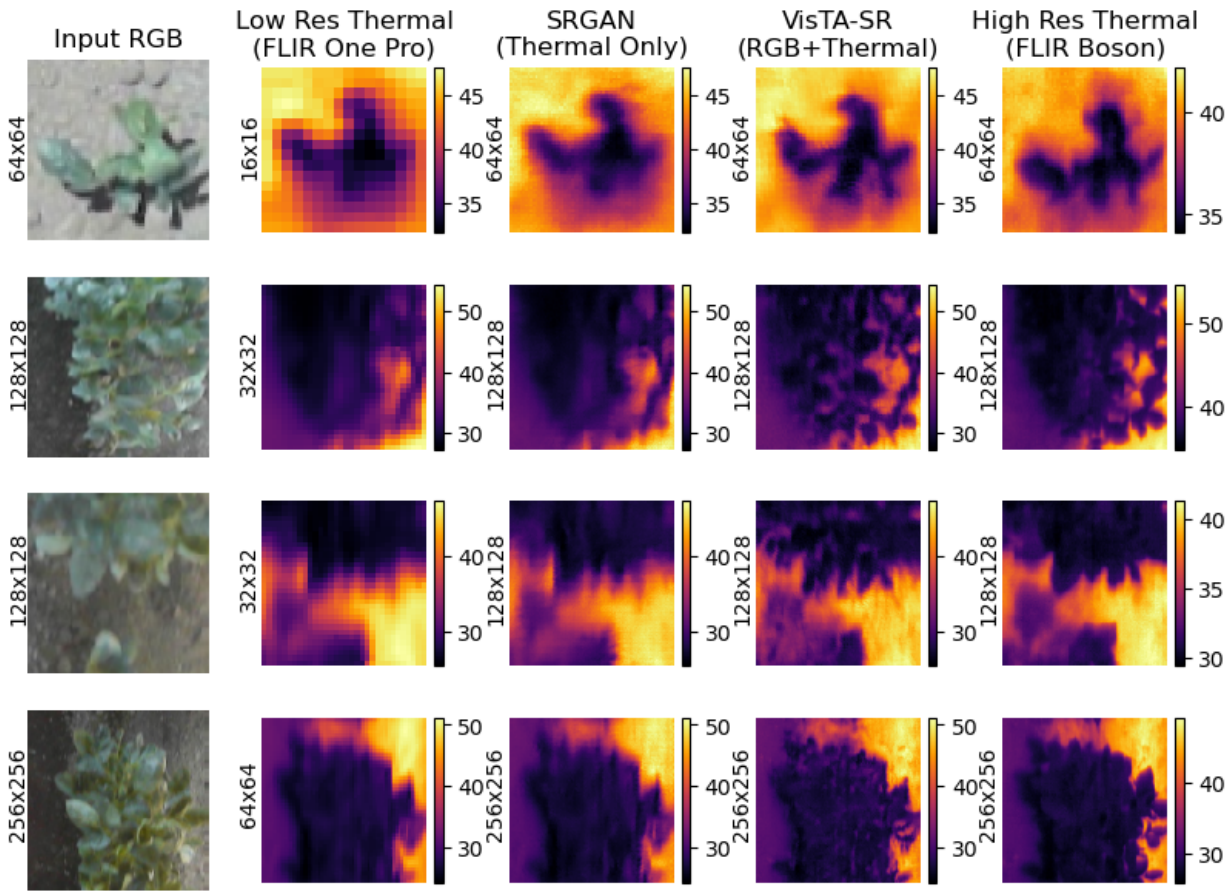


Figure 3.9. Comparison of input RGB, low-resolution thermal image input, SRGAN(Ledig et al., 2017) output in multiple image scales (64x64, 128x128, and 256x256), VisTA-SR output, and ground truth high-resolution thermal image

Table 3.4 compares the performance of Bilinear interpolation, Super-Resolution Generative Adversarial Network (SRGAN), and our proposed VisTA-SR algorithm. Bilinear interpolation yielded the highest RMSE but also the highest SSIM (Wang et al., 2004) and the lowest PSNR, while SRGAN and VisTA-SR demonstrated similar performance with an RMSE of 2.75 °C. It can be inferred that the higher RMSE value of the Bilinear algorithm is because SRGAN and VisTA-SR learned the temperature distribution of the training dataset, and Bilinear's higher SSIM value is believed to be a result of the original dataset already being aligned with the template matching process. SRGAN showed a higher PSNR value than VisTA-SR, but VisTA-SR exhibited better visual quality, indicating that evaluating super-resolution performance solely on image metrics is not ideal.

Table 3.4. RMSE, SSIM, and PSNR comparison of Bilinear, SRGAN, and VisTA-SR algorithms

Technique	RMSE (°C)	SSIM	PSNR
Bilinear	2.84	0.74	23.84
SRGAN (Wang et al., 2018)	2.74	0.63	24.26
VisTA-SR (Ours)	2.75	0.63	23.67

3.4 Conclusion and Future Work

This paper proposes a method to enhance temperature accuracy and image sharpness using a low-resolution thermal imaging camera for agricultural image acquisition. First, we conducted a calibration process to improve the temperature accuracy of the low-resolution thermal imaging camera, followed by field experiments for validation. It was confirmed that the temperature accuracy improved when using the calibrated parameters. We propose the VisTA-SR algorithm for converting low-resolution thermal images to high-resolution ones by aligning and combining RGB and low-resolution images. Through such improvements in temperature accuracy and image sharpness, we will be able to detect small temperature differences between crop tissues or parts, and analyze them in relation to genotypes, growth environments, growth stages, and various other factors.

One limitation was the difficulty of evaluating the performance of super-resolution algorithms in agricultural data using existing image metrics. Most super-resolution studies generate low-resolution images by down-sampling high-resolution images. In such cases, the pixels of low- and high-resolution pairs are perfectly aligned, making image metrics proportional to algorithm performance. However, in our study, low-resolution thermal images were collected separately from high-resolution images. Therefore, the output of our algorithm from low-resolution input may not perfectly match the high-resolution image at the pixel level. Given that image metrics change significantly with small pixel shifts, the evaluation metrics in Table 3.4 likely reflect errors from misalignment between camera systems,

even though VisTA-SR produced visually superior results. However, from an agricultural research perspective, temperature accuracy and the ability to detect plants are important for understanding their complex biophysical characteristics. In other words, developing specialized thermal image metrics for agricultural data that reflect these features for performance evaluation in future research is necessary. In future studies, we will examine whether the thermal image improvement algorithm maintains, improves, or hallucinates temperature information in thermal images. Also, using the thermal images processed with the algorithm developed in this paper, we will estimate biophysical parameters such as stomatal conductance in plants and compare accuracy with original and high-resolution image inputs.

Chapter 4

Vision Language Model for Organ-level Plant Architecture Generation from Simulated Images

Abstract

Three-dimensional (3D) procedural plant architecture models are used for simulation-based studies of plant structure and function, extracting plant architectural parameters from field measurements, and for generating realistic plants in computer graphics. However, measuring the architectural parameters for these models at the field and population scales remains prohibitively labor-intensive. We present a novel algorithm that generates the 3D plant architecture from an image, to create a functional-structural plant model that reflects organ-level geometric and topological parameters. Instead of using 3D sensors or processing multi-view images with computer vision to obtain the 3D structure of plants, we propose a method that generates token sequences containing a procedural definition of the plant architecture. This work uses only synthetic images for training and testing, where "exact" architectural parameters were known, which allowed for testing of the hypothesis that organ-level architectural parameters could be extracted from imagery data using a vision language model (VLM). A synthetic dataset of cowpea plant images was generated using the Helios 3D plant simulation library, with the detailed plant architecture encoded in XML files. We developed a plant architecture tokenizer for the XML file defining plant architecture, converting it into a token sequence that a language model can predict. Then,

a VLM was trained to predict plant architecture token sequences from images. Our results demonstrate that the model can predict plant architecture tokens with an F1 score (harmonic mean of precision and recall) (F1) score of 0.73 in a teacher-forcing method. Evaluation of the model was performed through autoregressive generation, achieving a bilingual evaluation understudy 4-gram (BLEU-4) score of 94.00% and a ROUGE-L score of 0.5182. Our model achieves lower mean absolute percentage error (MAPE) than feature regression-based methods in estimating bulk plant-level traits that require understanding of the occluded 3D structure of the plant, such as leaf count and leaf area. We conclude that generating plant architecture and parameter extraction from synthetic imagery are feasible using a VLM approach, which can be extended to real imagery in future work.

4.1 Introduction

Recent developments in vision language models (VLMs) that can understand images and reproduce structured output, such as comma-separated values (CSV), eXtensible Markup Language (XML), and 3D object files, have shown feasibility for generating structured output for 3D objects and shapes. For example, Wang et al. (2024b) introduced the LLaMA-Mesh algorithm, which is a vision language model that generates a sequence of natural language tokens defining 3D vertices and faces from an image. Since the methodology used in LLaMA-Mesh suggests the concept of generating language tokens defining 3D structure using conventional VLMs, it could be a potential solution that can reduce the time and cost of generating a structured plant architecture string. However, up until now, algorithms that use VLMs to generate structured output defining the plant architecture from images have not been tested in agricultural applications.

Therefore, this study addresses the following research questions: (1) Can VLMs accurately predict plant architecture token sequences from images? (2) How do different data and model configurations affect performance? The main contributions of this work are the de-

velopment of a tokenization method for plant architecture models and the validation of a hypothesis that a VLM can generate plant architecture models using advanced synthetic training data that incorporates foundational knowledge of plant architectures[1]. Model performance is assessed using sequence-based metrics (BLEU-4, ROUGE-L), architectural parameter fidelity, and plant bulk trait estimation accuracy. The use of synthetic training data not only provides an accurate benchmark for evaluating model training but also accelerates dataset generation, as obtaining such a benchmark is practically infeasible through real-world field experiments.[2] The tokenization method and synthetic dataset can support training larger VLMs for plant architecture generation.

4.2 Materials and Methods

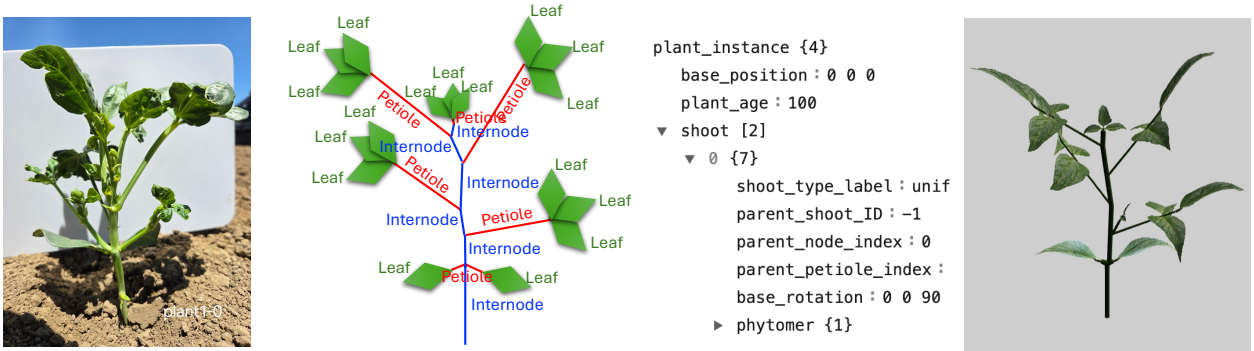
4.2.1 Plant Architecture Model

The typical structure of a dicotyledonous plant in vegetative growth consists of a shoot and a root system. This work focused on modeling the shoot system, which consists of internodes, petioles, and leaves. In reproductive growth, plants produce flowers, fruits, and seeds, changing the resource allocation and their branching pattern. In this paper, generating plants during vegetative growth was demonstrated, and plants in the reproductive stage will be addressed in future work. The basic plant architectural unit in the model is the phytomer, which consists of an internode with one or more petioles at its end, and each petiole contains one or more leaves (Teichmann and Muhr, 2015). Shoots are formed by connecting multiple phytomers end-to-end, which occurs when the vegetative bud at the shoot tip develops into a new phytomer (i.e., "growth"). Shoots can spawn lateral child shoots from vegetative buds, which are located where the internode and petiole are attached, creating a nested topological structure. For example, Fig. 4.1 (a) shows an actual cowpea plant, and Fig. 4.1 (b) represents the corresponding topological diagram. Knowing

[1]Code is available here: https://github.com/GEMINI-Breeding/Image2PlantArchitecture
[2]Dataset is available here: https://huggingface.co/datasets/heesup/Cowpea-Architecture-XML

the order and relationship among plant organs allows us to implement the actual 3D plant structure in a simulation program.

In this work, the Helios plant simulation library was used to generate randomly varying cowpea plants based on its architectural model. Bailey (2019) developed Helios, a scalable 3D plant and environmental biophysical modeling framework. This library is implemented in C++ and incorporates sub-models, known as plugins. Helios can simulate various types of plant biophysical processes, including energy balance, photosynthetically active radiation (PAR), and photosynthesis simulations, based on 3D geometry and ray tracing. It is possible to simulate plant structure and growth and output the full plant structure in XML file format using the plant architecture plug-in.



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

Figure 4.1. Schematic depiction of how model plant architecture is defined from real plants. (a) Cowpea plant in its early growth stage. (b) Simplified diagram with plant organ annotations. (c) Example plant architecture XML created based on the annotations. (d) Image rendering of the plant architecture XML using the Helios simulator.

The plant architecture can be encoded in an XML file format, which uniquely parameterizes the topology and geometry of plant organs. The XML file contains two types of plant architectural information: the relative connectivity of plant organs (i.e., topology) and the parameters defining the geometry of each plant organ. The relative connectivity of plant organs is defined by the order in which the plant organs are defined in the XML syntax and the ID of the parent instance. For example, the parameter set defining a shoot has the

shoot's unique ID and information about the parent node where the shoot is located. The parameters defining the shape of each plant organ include length, diameter, scale, angle, etc., depending on the type of plant organ, and define the individual geometric characteristics of the plant organ. Table 4.1 summarizes the plant architectural parameters used in the Helios plant architecture XML structure. The XML files include base positions and plant age as comprehensive plant-level parameters. The parameters defining a shoot encompass the shoot type along with its rotational parameters. The internode parameters provide geometric information, such as length, radius, pitch, and phyllotactic angle. Connecting the leaf to the internode, the petiole is characterized by parameters that specify its length, radius, pitch, curvature, and scale. Lastly, each leaf is associated with a scale parameter that determines its size, in addition to pitch, yaw, and roll angles, which indicate its angle transformation from the parent organ's coordinate system. Collectively, these elements provide a comprehensive representation of the plant's structure and are defined for each geometric element.

There are two approaches to generating the architecture of a plant via an XML file. The first approach involves manually creating the plant structure and writing it directly to an XML file, which can then be read into Helios to generate the plant geometry. Fig. 4.1 (c) is an example of a plant architecture XML file based on a real plant, and Fig. 4.1 (d) shows the rendering of this using the Helios program. The second way is to generate the plant architecture dynamically and procedurally based on the Helios plant architecture plug-in. The plug-in can automatically generate a plant architecture model of a specific plant growth stage, simulate plant growth, and modify the plant architecture model by pre-defined rules and random sampling from the parameter distribution of plant organs.

Table 4.1. List of plant architectural parameters from Helios plant architecture plug-in. The default values and parameter distributions are used from Helios Cowpea library. More detailed information can be found at https://baileylab.ucdavis.edu/software/helios

Parameter	Description	Unifoliate	Trifoliate	Unit
base_rotation_pitch	Plant pitch	Uniform(0, 10)		deg
base_rotation_yaw	Plant yaw	Uniform(0, 360)		deg
base_rotation_roll	Plant roll	Uniform(0, 360)		deg
plant_age	Days after planting	Uniform(0, 39)		days
shoot_type_label	Shoot types	Constant(1)	Constant(3)	index
shoot_base_pitch	Shoot base pitch angle	Constant(40)	Uniform(40, 60)	deg
shoot_base_yaw	Shoot base yaw angle	Uniform(0, 360)	Uniform(-20, 20)	deg
shoot_base_roll	Shoot base roll angle	Constant(90)		deg
internode_length	Initial internode length	Constant(0.002)		m
internode_radius	Initial internode radius	Constant(0.0015)		m
internode_pitch	Internode pitch angle	Constant(0)	Constant(20)	deg
internode_phyllotactic_angle	Phyllotactic angle	Uniform(145, 215)		deg
petiole_length	Petiole length	Constant(0.0004)	Uniform(0.06, 0.08)	m
petiole_radius	Petiole radius	Constant(0.0001)	Constant(0.0018)	m
petiole_pitch	Petiole pitch angle	Uniform(60, 80)	Uniform(45, 60)	deg
petiole_curvature	Petiole curvature	N/A	Uniform(-200, -50)	deg
leaflet_scale	Leaflet size scale	N/A	Constant(0.9)	–
leaf_scale	Leaf size scale	Constant(0.02)	Uniform(0.09, 0.12)	–
leaf_pitch	Leaf pitch angle	Uniform(-10, 10)	Normal(45, 20)	deg
leaf_yaw	Leaf yaw angle	Constant(0)	Constant(10)	deg
leaf_roll	Leaf roll angle	Constant(-15)		deg

4.2.2 Synthetic Cowpea Dataset

It is necessary to have ground truth annotations of the plant architecture corresponding to the plant images to predict the plant architecture from images using VLMs. However, acquiring a large-scale agricultural image dataset with high-quality annotations is costly and time-consuming, which can limit algorithm development (Dyrstad and Øye, 2025). To overcome these difficulties, synthetic data generated by simulation programs can be utilized.

A dataset consisting of 3D cowpea plant models was procedurally generated based on randomly varying architectural parameters following a specified distribution (Table 4.1). The dataset was generated using the procedural plant architecture model in Helios v1.3.14, which is described in more detail in the previous section. During the dataset generation process, consistent random seeds were assigned to each plant, based on the generated plant index for reproducibility (Fig. 4.2 (a)). The growth of the crop was simulated by creating a plant on day 0 using the Helios Cowpea library and then advancing it over time to day 39 (Fig. 4.2 (b) and (c)). All simulated cowpea plants in this dataset were in the vegetative growth stage, simplifying the problem definition by generating a plant architecture model without reproductive organs. The distributions of plant organ parameters are defined within the Helios library and can be customized by the user. For the generation of this dataset, the default parameter distributions defined in the library were used. As a result, the generated plant architecture dataset represents an imaginary population of cowpea plants that were sampled from specific parameter distributions. 10,000 random seeds were used to generate the day 0 plant, and plant growth was simulated on a daily basis up to day 39. Therefore, a total of $10{,}000 \times 40 = 400{,}000$ simulated images and plant architecture XML pairs were generated. For multi-view dataset generation, images at viewing angles of 0°, 120°, and 240° were also rendered for each XML file (Fig. 4.2 (d)).

Figure 4.2. Process of generating the synthetic cowpea plant architecture dataset. (a) Random seeds ensure reproducibility. (b) Plant architecture is procedurally generated using the Helios Cowpea library. (c) Growth simulation from day 0 (seedling) to day 39 (vegetative growth). (d) Optional multi-view rendering at azimuthal angles of 0°, 120°, and 240°.

4.2.3 Plant Architecture Tokenization

Helios can encode and decode XML files (Fig. 4.1 (c)), a structured document format that enables the export or import of plant architecture models to use with Helios. While this text-based format supports VLM training, processing XML files requires an intermediate step that converts the structured text into embedding vectors from which the model can learn. However, applying a conventional text tokenization method to an XML text is inefficient because the XML file structure and elements, such as opening and closing tags with special characters, unnecessarily increase the sequence length that requires a larger context length for language models, resulting in a larger memory footprint and reduced generation performance. Therefore, a new method was needed to streamline the XML file to essential

plant architecture model information. In this paper, a specialized plant architecture tokenizer was developed that represents the plant organs and parameters as tokens.

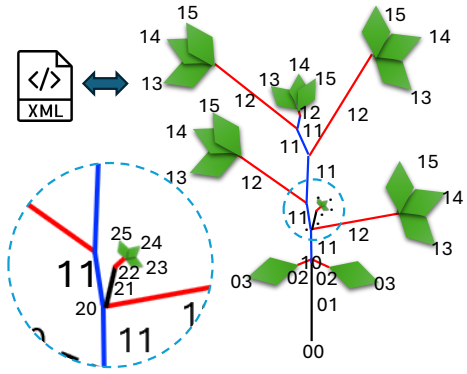
The XML file contains information on a series of plant organs with a hierarchical structure, which can be represented by their organ token and the following parameter tokens using the plant architecture tokenizer. The plant organ tokens include the branching order in which the organ is attached, as well as the type of organ. This preserves the hierarchy and sequential relationship between plant organs and makes the plant architecture token sequence semantically identical with the original XML file, which enables conversion between the two formats without information loss.

Fig. 4.3 illustrates an example of tokenizing plant architecture into a sequence of tokens. The first shoot tag is tokenized as "00", where the leading 0 indicates the primary branch, and the trailing 0 represents the shoot organ. Next, the phytomer structures of internode, petiole, and leaves appear repeatedly. Similarly, the trailing number represents 1: internode, 2: petiole, 3: leaf. Therefore, the internode connected to the primary branch is tokenized as "01," and the petiole attached to this internode is tokenized as "02". Since the initial leaves connected to the primary branch are unifoliate, each leaf is attached to a single petiole, which is tokenized to "03". The shoot organ corresponding to the secondary branch connected to internode "01" is tokenized as "10", and the internode and petiole connected to this node are tokenized as "11" and "12", respectively. Since the leaves attached to the petioles after the first unifoliate leaves have a trifoliate configuration, each leaf is tokenized as "13", "14", and "15". Parameter tokens defining the plant organ's geometry follow the plant organ token, represented as p0 through p3 or p4, depending on the number of parameters per plant organ. This rule is applied recursively until all the plant organs are described. Combining all of this together, the plant architecture defined in Fig. 4.3(a) can be tokenized as shown in Fig. 4.3(d).

After converting the plant architecture XML to a token sequence, it is converted to cat-

egorical variables, which are token IDs that the transformer can predict, using a table shown in Fig. 4.3 (c). First, for the plant organ token, the branching order was multiplied by 6, and the organ type was added to make the token IDs. Next, the parameter values are converted to the token IDs. Because conventional tokenizers do not have specific token dictionaries for decimal point values, they can't convert the plant parameter token value into a single token ID. Therefore, a quantization method was applied to convert the parameter values to discrete integer values between 24 and 222, considering varying scales and distributions. To ensure a similar resolution across various scales and reduce the size of token dictionaries, combinations of a set of constant values and linearly spaced float grids were applied. For example, 0, -10, 10, and 90 degrees are frequently observed numbers in angle parameters. Also, 1 and 3 are used to represent the shoot organ that has unifoliate leaves or trifoliate leaves. As a result, those constant values are added to the mapping between parameter token values to token IDs. Besides constant values, linearly spaced grids were applied. This includes angle representations of [-40, 360] in degrees in 2.5-degree resolution. The set of decimal values was created by defining 10 evenly spaced samples within the ranges [0.1, 1.0], [0.01, 0.1], [0.001, 0.01], and [0.0001, 0.001] in meters to represent organ parameters with small decimal point values. This set of constant values and float value grids was concatenated and used to find the closest index of a certain parameter value, which can be mapped to a token ID.

(a) Example plant organ labels



(b) Plant token annotation rules

Format: MN p0 p1 p2 p3 p4

Branching order (M)
0: Primary branch
1: Secondary branch
2: Tertiary branch

Organ (N)
0: shoot
1: internode
2: petiole
3, 4, 5: leaf

Parameters
p0 ... p4: organ parameters

(c) *Token IDs*

Branching order (M)	Organ (N)	IDs (6*M + N)
0 - 3	0 - 5	0 - 23
Parameters		**IDs**
Quantized floats values		24 - 222
Special Tokens		**IDs**
SOS		223
META		224
PAD		225
EOS		226

(d) *Tokenized plant architecture (1x201)* * ***Bold**: Organ tokens and special tokens / Plain: Parameter tokens*

00,p0,p1,p2,p3,p4,**01**,p0,p1,p2,p3,**02**,p0,p1,p2,p3,p4,**03**,p0,p1,p2,p3,**10**,p0,p1,p2,p3,p4,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**20**,p0,p1,p2,p3,p4,**21**,p0,p1,p2,p3,**22**,p0,p1,p2,p3,p4,**23**,p0,p1,p2,p3,**24**,p0,p1,p2,p3,**25**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**02**,p0,p1,p2,p3,p4,**03**,p0,p1,p2,p3

(e) Plant metadata and *special tokens added (1x208)*

SOS,**META**,p0,p1,p2,**META**,**00**,p0,p1,p2,p3,p4,**01**,p0,p1,p2,p3,**02**,p0,p1,p2,p3,p4,**03**,p0,p1,p2,p3,**10**,p0,p1,p2,p3,p4,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**20**,p0,p1,p2,p3,p4,**21**,p0,p1,p2,p3,**22**,p0,p1,p2,p3,p4,**23**,p0,p1,p2,p3,**24**,p0,p1,p2,p3,**25**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**11**,p0,p1,p2,p3,**12**,p0,p1,p2,p3,p4,**13**,p0,p1,p2,p3,**14**,p0,p1,p2,p3,**15**,p0,p1,p2,p3,**02**,p0,p1,p2,p3,p4,**03**,p0,p1,p2,p3,**EOS**

(f) *Token IDs (1x208)*

223,*224*,*72,60,62,224*,***0***,*40,40,114,82,77,****1****,51,41,40,139,****2****,43,41,107,133,77,****3****,52,80,40,34,****6****,40,40,40,82,79,****7****,50,41,80,40,****8****,52,40,106,118,76,****9****,51,39,82,34,****10****,52,81,40,34,****11****,51,79,36,34,****12****,90,40,40,82,79,****13****,45,40,40,40,****14****,43,40,40,118,76,****15****,45,39,82,34,****16****,45,81,40,34,****17****,45,79,36,34,****7****,50,41,40,150,****8****,52,40,96,118,76,****9****,51,39,82,34,****10****,52,81,40,34,****11****,51,79,36,34,****7****,50,41,36,150,****8****,52,40,100,118,76,****9****,51,39,82,34,****10****,52,81,40,34,****11****,51,79,36,34,****7****,50,41,40,150,****8****,52,40,100,118,76,****9****,51,39,82,34,****10****,52,81,40,34,****11****,51,79,36,34,****7****,45,41,82,24,****8****,41,40,100,118,76,****9****,50,39,82,34,****10****,50,81,40,34,****11****,50,79,36,34,****2****,43,41,107,133,77,****3****,52,38,40,34,****226***

Figure 4.3. An example of plant architecture annotations and tokenization. (a) shows a simplified plant architecture diagram based on an input XML file and (b) plant organ annotation rules. (c) shows the rules converting plant organ annotations and float values of plant organ parameters to unique integer values. (d) shows plant architecture tokens of example plant architecture, which can also be represented in XML file format. (e) shows the sequence with special tokens for VLM training. (f) shows the token IDs that the model predicts as categorical variables.

As cropping and resizing the input image during preprocessing results in the loss of plant size information, the plant's width and height in meters were estimated from the original image and concatenated in the input token string. The vegetation fraction, the proportion of the image window covered by vegetation, was calculated and included as plant meta-

data. This plant metadata was encapsulated with <META> tokens and added at the start of the plant architecture tokens. Next, a <SOS> token indicating the beginning of the sentence and an <EOS> token indicating the end of the sentence were added to the beginning and end of the sequence. Fig. 4.3(e) shows an example with plant metadata tokens, including start and end tokens, and Fig. 4.3(f) shows the final input token IDs, ready for training the language model head.

4.2.4 Algorithm Structure

After tokenizing the plant architecture XML into a token sequence, the model was trained to perform the next token prediction (NTP) task. A teacher-forcing method is a training strategy in which the ground truth token is provided as input at each decoding step rather than the model's prediction (Williams and Zipser, 1989). Fig. 4.4 shows the overall algorithm structure. As mentioned in the previous section, the input image was cropped around the plant and resized to 224^2 or 448^2 based on the model configuration. The plant size information was also quantized and concatenated to the beginning of the input token IDs. The NTP tasks estimated the next token starting from the SOS token and plant metadata tokens to complete the plant architecture token sequence. The target token sequence was one token shifted to predict the start of the metadata token to the <EOS> token.

The model and training code were implemented in Python programming language based on the Hugging Face (https://huggingface.co) library's vision encoder-decoder model, which consists of layers of transformer encoder and decoder blocks (Dosovitskiy et al., 2021; Vaswani et al., 2017). A vision foundation model DINOv2 (Oquab et al., 2024) was utilized with its weights frozen to serve as a feature extractor, thereby maintaining generalization performance and reducing training time. The vision encoder's patch embedding sequence was connected to the decoder's cross-attention layer to perform the NTP task with a vision context. The decoder block's configuration follows the GPT-2 model series.

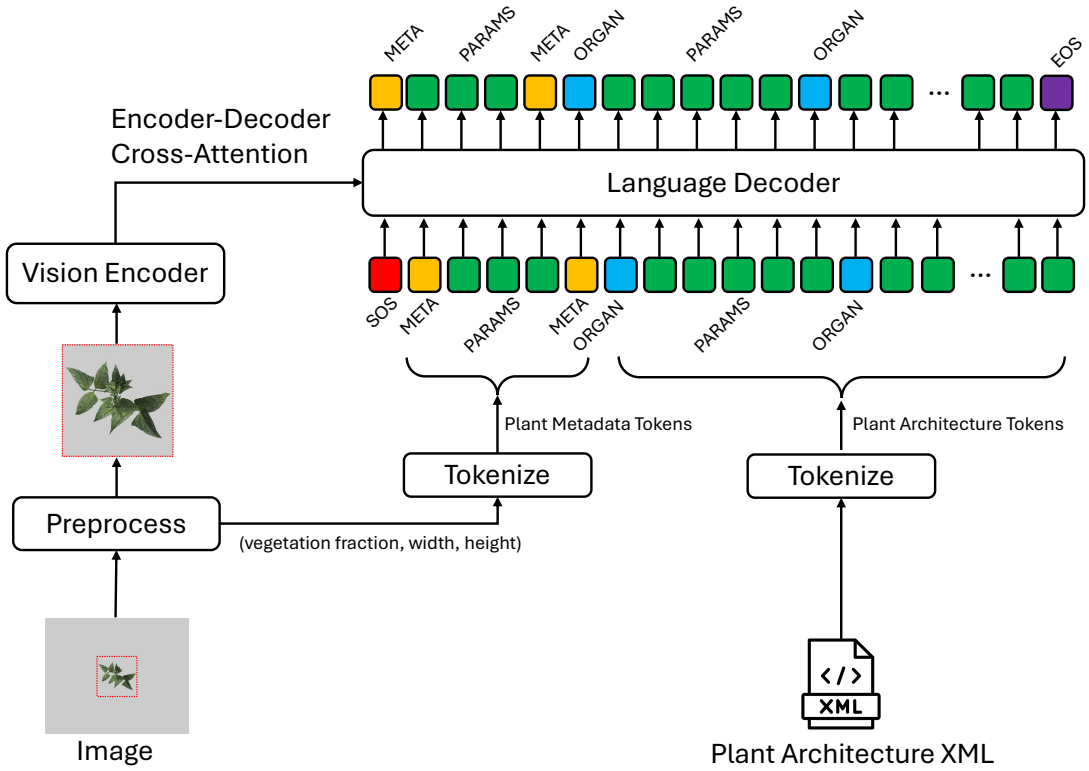


Figure 4.4. Proposed algorithm structure for the next token prediction task from an image. A plant architecture XML was visualized by a plant simulator to create an input image for the model and tokenized to provide a ground truth token sequence. The preprocess block cropped the input image around the plant, resized it to the vision encoder input size, and saved the original plant size information, which is plant metadata. The tokenize block for plant metadata converts plant metadata to input token IDs and concatenates them after the SOS token (red). The vision encoder block extracted visual features from the preprocessed image and was connected to the encoder-decoder cross-attention layer of the language model head. The target sequence was one token shifted, and the <EOS> token (purple) was added at the end of the sequence to train the language model head in a teacher-forcing method.

4.2.5 Model Training Procedure

To examine the performance differences based on the number of parameters in the vision encoder and language decoder within the vision encoder-decoder model, two combinations of vision encoders (ViT-S, 21M parameters; and ViT-B, 86M parameters) and two language decoders (gpt-2, 115M parameters; and gpt-2-medium, 406M parameters) were tested. To understand how the amount of information affects plant architecture generation performance, training was conducted using both top-view images and multi-view images, which include side-view images, with combined input images resized to 224 and 448 pixels

in width and height for the training dataset. Consequently, a total of 16 different model combinations were trained, and their evaluation metrics were calculated.

During the training process, 80% of the dataset was used as training data, 10% as validation data, and 10% was reserved as a test dataset. Splits were performed at the plant level: all 40 daily time points and all three viewing angles (0°, 120°, 240°) from each plant were kept together in the same split, preventing data leakage across sets. To increase stability in the early stages of training (Hägele et al., 2024), the learning rate was gradually increased from 0 to 10^{-4}, reaching 20% of the total training steps, and then linearly decreased to 0 for the remaining steps to complete the training. The training batch size was 16, and the gradient accumulation size was set to 4, resulting in an effective batch size of 64. Training was conducted to minimize the cross-entropy loss function and was carried out on an NVIDIA A100 graphics processing unit (GPU) for 4 epochs.

The accuracy and weighted F1 score were monitored in a teacher-forcing method during the model training, where the model estimates the n^{th} token given the previous true $(n-1)^{th}$ tokens. The training accuracy of plant architecture tokens was calculated using the numbers of true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP) between the predicted n^{th} token from 1 to $(n-1)^{th}$ token sequence and the true n^{th} token.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{4.1}$$

The weighted F1 score was calculated as the harmonic mean of precision and recall weighted by the percentage of token IDs. For a given token k, the precision, recall, and

F1 score are defined as:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \tag{4.2}$$

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{4.3}$$

$$\text{F1}_k = \frac{2 \times \text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{4.4}$$

The final Weighted F1 score was computed as:

$$\text{Weighted F1} = \frac{1}{\sum_{k=0}^{227} n_k} \sum_{k=0}^{227} n_k \text{F1}_k \tag{4.5}$$

4.2.6 Model Evaluation Metrics

In the model evaluation stage, the trained model generates plant architecture tokens from given SOS and plant metadata tokens with image context, also known as autoregressive generation. Fig. 4.5 shows the overall evaluation process. As shown in Fig. 4.5 (b), the predicted token was then appended to the input token sequence, and this process continued until the <EOS> token was generated, thereby completing the plant architecture token sequence. A beam search with a beam size of 6 was employed to improve the generation result.

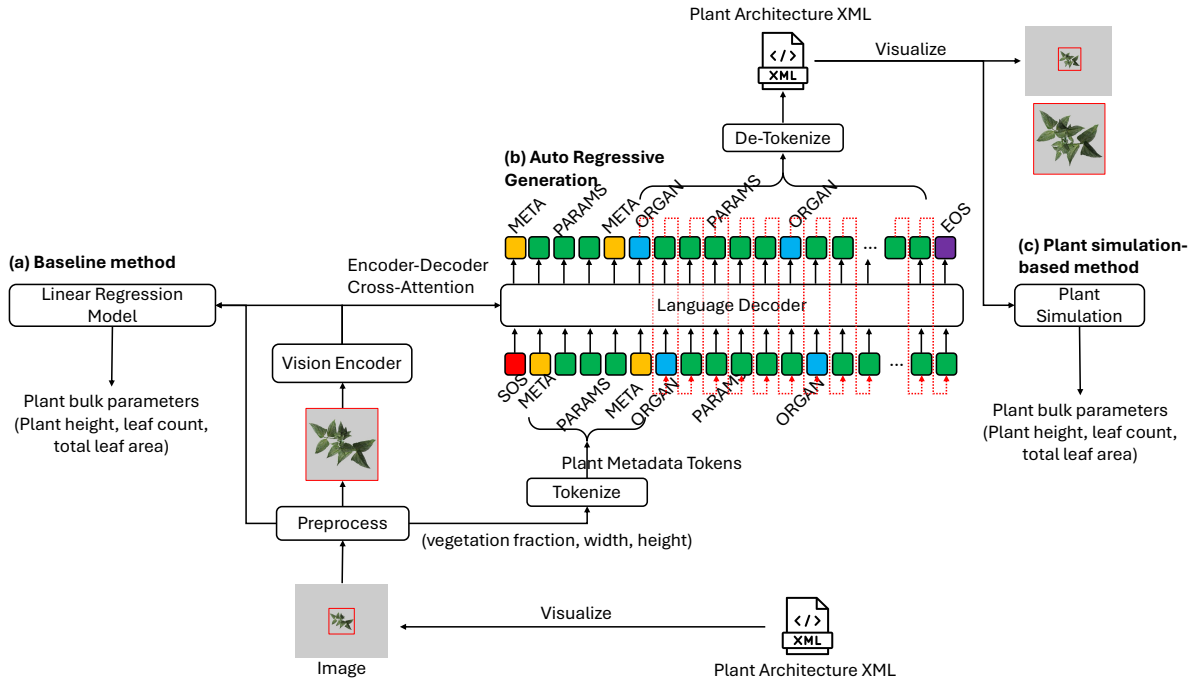


Figure 4.5. Visualization of the model evaluation process. (a) Baseline method based on linear regression: image features and plant metadata values are combined to predict plant traits. (b) In autoregressive generation (red dotted arrows), each generated token is appended to the input sequence until the <EOS> token is produced. (c) The generated plant architecture XML is visualized for qualitative evaluation and loaded into a plant simulation program to compute plant structural traits.

Unlike the teacher-forcing training, the generation results have varying lengths from the ground truth, resulting in different numbers of plant organs. Therefore, it is necessary to use evaluation metrics for comparing two sequences of different lengths. To do this, the BLEU-4 score (Papineni et al., 2001) and ROUGE-L (Lin, 2004) were calculated.

The bilingual evaluation understudy (BLEU) score is a precision-based metric comparing token sequences between the generated and ground truth. Even if the metric is developed for machine translation evaluation, it can be extended to compare any two token sequences. The BLEU score counts how many n-gram patterns in a generated token sequence also appear in the ground truth token sequence. The N-grams are groups of N consecutive tokens within a token sequence. The 1-gram BLEU score calculates the ratio of tokens in the generated token sequence that are also present in the ground truth. Since plant organ tokens (i.e., "00", "01", "02", ... "35") appear at least once in any generated token

sequence, the ratio of plant organ tokens in the generated sequence, approximately 17% to 20%, serves as a regression-based baseline for the 1-gram matching score if the generated token sequence contains all the plant organ tokens but wrong plant parameter tokens. Similarly, if a correct combination of plant organ tokens with three organ parameter tokens, or any consecutive combinations of four tokens, also appear in ground truth, it is counted as a 4-gram matching.

The BLEU-4 score is calculated by averaging 1-gram, 2-gram, 3-gram, and 4-gram matching scores. A higher BLEU-4 score implies that the generated tokens have common patterns with the ground truth. In other words, a lower BLEU-4 score means that the generated token sequence has patterns that do not exist in the ground truth. For example, if the BLEU-4 score is 100%, it indicates a sufficient condition that all the patterns of the generated plant architecture tokens are present in the ground truth plant architecture token sequence. It also means that if the model generated only one branch with correct patterns, it could still show a high precision score. To prevent this, a brevity penalty is applied to penalize the sequence length of the generation result if it's shorter than the ground truth.

$$BLEU4 = \min\left(1, \exp\left(1 - \frac{\sum l_{GT}}{\sum l_{gen}}\right)\right)\left(\prod_{n=1}^{4} \text{n-gram BLEU}\right)^{1/4} \tag{4.6}$$

ROUGE-L calculates the longest common subsequence (LCS) between the generated token sequence and the ground truth token sequence and uses this to compute the F score. The F_{LCS} score for the i^{th} sentence is calculated as follows:

$$R_{LCS} = \frac{\text{LCS}(\text{Gen}, \text{GT})}{\text{len}(\text{Gen})} \tag{4.7}$$

$$P_{LCS} = \frac{\text{LCS}(\text{Gen}, \text{GT})}{\text{len}(\text{GT})} \tag{4.8}$$

$$F_{LCS} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta P_{LCS}}, \quad \text{where } \beta = \frac{P_{LCS}}{R_{LCS}} \tag{4.9}$$

The ROUGE-L score is the average of the F_{LCS} scores for N pairs of generated plant architecture token sequences and their corresponding ground truth plant architecture token sequences. A higher ROUGE-L score indicates that longer common token patterns are found between the generated plant architecture token sequence and the ground truth plant architecture token sequence. If the ROUGE-L score is 1.0, it means that the generated plant architecture token sequence is the same as the ground truth plant architecture token sequence. Even if the ROUGE-L score can show low scores for inverted sequences (i.e., same plant architecture with a swapped secondary branch) and paraphrased sequences (i.e., visually similar plant architecture but has negligible token offsets for every organ parameter), the score can be a metric that shows structural similarity and preservation of pivotal plant architecture patterns.

The evaluation used 4,000 images from the dataset, which were not used during the training or validation step. The evaluation step was done after finishing the model training and validation for each model configuration.

4.2.7 Plant Architecture Parameter Evaluations

After the model evaluation, the best-performing model that achieved the highest BLEU-4 and ROUGE-L scores, as indicated in Table 4.2, was selected for a detailed evaluation in terms of plant architecture. First, we compared the tokens related to the plant's base rotation angles and plant age from the generated and the ground truth plant architectures.

The base rotation angles are the primary branch's rotation angles, which determine the plant's overall rotation and other plant organs attached to it. The plant age indicates the number of days since germination in the simulation. In addition to the direct parameter comparison above, the distributions of the other organ parameters were compared to determine whether the generated plant architecture organ parameters followed the distribution of the ground truth. The parameter distributions were plotted as histograms, and the Wasserstein distance (WD) was computed to measure the differences between the two distributions. WD is a method for measuring the distance between two probability distributions, defined as follows:

$$W_p(P_r, P_g) = \left(\mathbb{E}_{(x,y)\sim\gamma}[d(x,y)^p]\right)^{1/p} \tag{4.10}$$

When p = 1, it is referred to as the earth mover's distance, and a smaller value indicates that the two distributions can be matched at a lower cost.

4.2.8 Plant Simulation-Based Trait Evaluation

Plant height, leaf count, leaf area, and leaf angle distribution are important factors that influence the photosynthetic and water-use efficiency of plants, also affecting yield and environmental resistance. For instance, Truong et al. (2015) used functional structural plant model (FSPM) to demonstrate that differences in leaf angles can affect the amount of photosynthesis, which in turn can influence the yield of sorghum. Pele et al. (2016) reported that plant height, number of leaves per plant, stem diameter, and root dry mass of cowpea affect the drought susceptibility index.

Calculating the number of plant organs, such as leaf count and number of nodes, can be done by analyzing the plant architecture sequence. However, estimating plant architectural traits such as plant height and leaf angle distribution directly from the plant architecture sequence is not possible because tokens are the basic building blocks of plant structure,

not a simulated plant model. The angle of each leaf is influenced by the series of coordinate transforms defined by parameter tokens of the internode and petiole from the first shoot to the leaf. Similarly, plant height is determined by a series of coordinate transforms and measured by the highest point above the ground. Therefore, the plant architecture sequences were loaded into a 3D plant model using a simulation program, and the final values were computed afterward Fig. 4.5 (c).

A linear regression-based baseline method was tested to evaluate the accuracy of plant height, leaf count, and total leaf area that were indirectly calculated through plant architecture model simulation. The baseline method for predicting plant traits from an image was fitting linear regression models from image features and plant metadata Fig. 4.5 (a). While the regression models cannot predict the entire complex plant architecture token sequence, traits that represent a plant's overall visual traits, such as plant height, leaf count, and leaf area, can be predicted with relatively high accuracy (Schiller et al., 2021; Singh et al., 2023; Zheng et al., 2022). Therefore, linear regression models based on fully connected layers, using the ViT-B model as a feature extractor, were fitted to predict plant height, leaf count, and total leaf area as a baseline.

4.3 Results

4.3.1 Model Training and Evaluation Results

A total of 16 combinations of settings were trained, including view information (top-view vs. multi-view), input image size (224^2 vs. 448^2 pixels), size of vision encoder model (ViT-S vs. ViT-B), and size of text decoder model (gpt-2 vs. gpt-2-medium). Table 4.2 summarizes the training metrics, showing the accuracy and weighted F1 score. Overall, both the weighted F1 scores showed values ranging from 0.72 to 0.73. In particular, the weighted F1 score and accuracy showed similar results regardless of the model size and input images. This indicates that, given the current data and model architecture under the teacher forc-

Table 4.2. Training metrics for models with different encoder and decoder setups, across various viewing angles and input image sizes. The highest F1 score and accuracy are shown in **bold**.

			Text Decoder			
			gpt-2		gpt-2-medium	
View	**Input Size**	**Vision Encoder**	**Accuracy**	**F1**	**Accuracy**	**F1**
Top-view	224^2	ViT-S	0.7277	0.7285	0.7282	0.7290
		ViT-B	0.7260	0.7271	0.7291	0.7309
	448^2	ViT-S	0.7269	0.7271	**0.7301**	0.7317
		ViT-B	0.7276	0.7286	0.7284	0.7294
Multi-view	224^2	ViT-S	0.7270	0.7278	0.7287	0.7288
		ViT-B	0.7297	**0.7329**	0.7288	0.7290
	448^2	ViT-S	0.7286	0.7292	0.7297	0.7306
		ViT-B	0.7283	0.7296	0.7290	0.7303

ing training method, the maximum token prediction accuracy was approximately 0.73.

Table 4.3 compares autoregressive generation performance across model architectures and input configurations. Among the 16 configurations tested, the multi-view input with 448^2 resolution, ViT-B encoder, and gpt-2-medium decoder achieved the highest BLEU-4 score (94.00%) and ROUGE-L score (0.5182). The gpt-2-medium decoder consistently outperformed the smaller gpt-2 decoder across all vision encoder and input size combinations. Image resolution showed variable effects: while 448^2 inputs generally improved ROUGE-L scores, BLEU-4 scores did not consistently increase with resolution. Multi-view inputs outperformed single top-view inputs in 6 of 8 comparable configurations for BLEU-4, and in all configurations for ROUGE-L.

Fig. 4.6 shows rendered images of a plant architecture sequence generated by the best-performing model (multi-view, 448, ViT-B, and gpt-2-medium), after converting the generated plant architecture token sequence from the input image back into XML and rendering it.

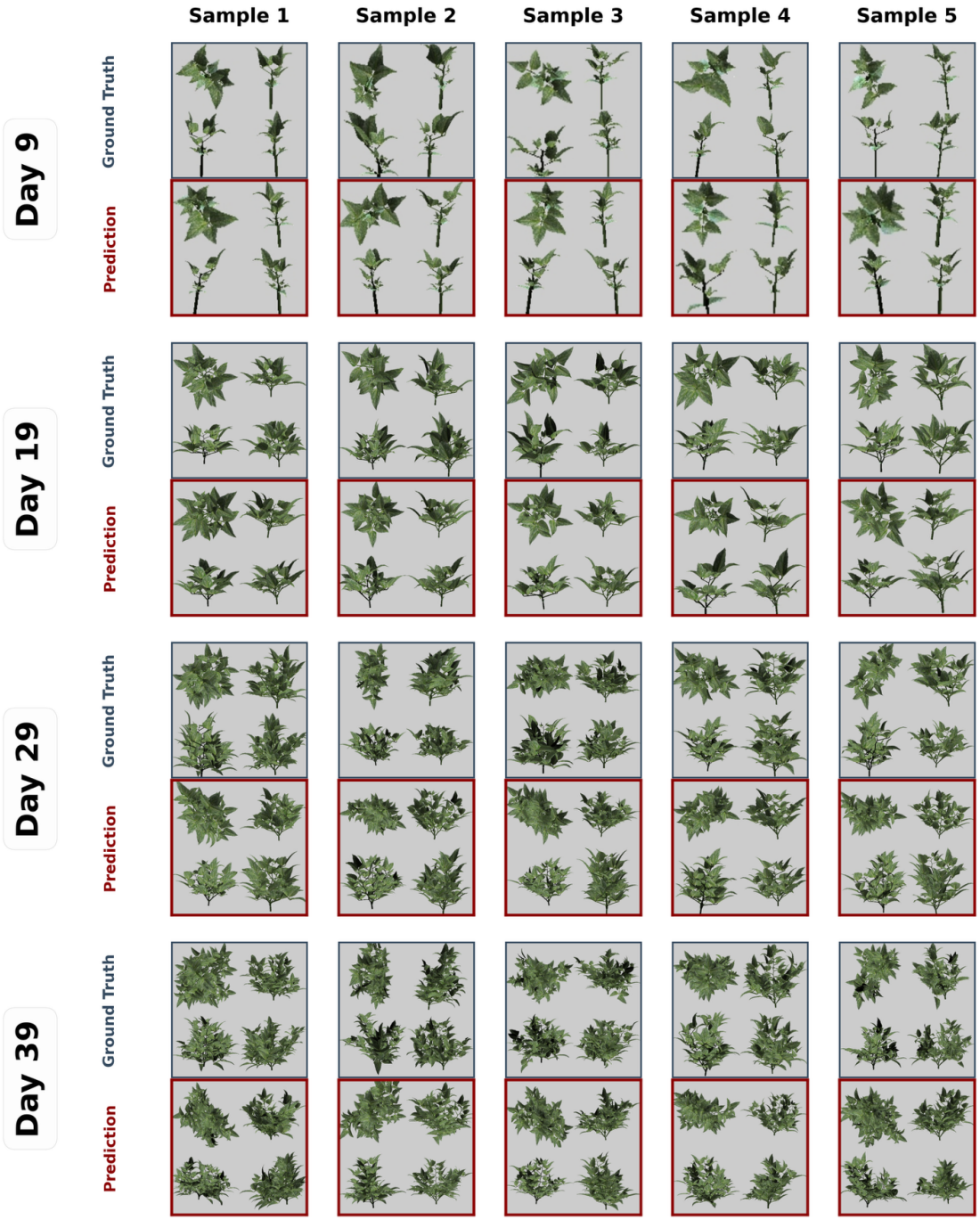


Figure 4.6. Examples of rendered images of plant architecture tokens generated from the best-performing model (multi-view, 448, ViT-B, and gpt-2-medium), showing the highest BLEU-4 and ROUGE-L scores.

Table 4.3. Model evaluation metrics for different configurations. The highest BLEU-4 and ROUGE-L scores are shown in **bold**.

			Text Decoder			
			gpt-2		gpt-2-medium	
View	**Input Size**	**Vision Encoder**	**BLEU-4 (%)**	**ROUGE-L**	**BLEU-4 (%)**	**ROUGE-L**
Top-view	224^2	ViT-S	92.35	0.4998	90.67	0.5056
		ViT-B	87.07	0.5023	87.85	0.5062
	448^2	ViT-S	84.97	0.4996	85.34	0.5070
		ViT-B	90.73	0.5030	90.79	0.5082
Multi-view	224^2	ViT-S	89.52	0.5093	91.10	0.5154
		ViT-B	84.88	0.5112	87.96	0.5150
	448^2	ViT-S	85.39	0.5090	84.27	0.5175
		ViT-B	89.08	0.5126	**94.00**	**0.5182**

4.3.2 Generated Plant Architecture Token Sequence Evaluations

To examine how precisely the model generates the plant architecture parameters, the pitch, yaw, and roll angles of the primary branch were compared between the generated plant architecture token sequence and the ground truth. Due to the conversion from parameter to token IDs, which converts float values into discrete integers, the parameter values exist in clusters. For example, in Fig. 4.7 (a), the base rotation pitch sampled from a [0, 10] uniform distribution is quantized to discrete float values ranging from 0 to 1.0 and 2.5-degree angle steps, resulting in a grid-like scatter plot. The discrete patterns introduced by tokenization in yaw and roll angle of the primary branch, which range from 0 to 360 degrees, were less pronounced Fig. 4.7 (b) and (c). The range of pitch angle of the primary branch was narrow ([0, 10]) in the synthetic dataset and showed a subtle effect on the image, making its prediction challenging. The yaw and roll angles of the primary branch were closely clustered to the y=x line; however, there are clusters observed in y=x-180 and y=x+180, suggesting the model has difficulty with 180-degree rotations around the element's roll axis. Fig. 4.7 (d) compares the plant age from the generated plant architecture token with the plant age from the ground truth. The plant age prediction showed an R^2 value of 1.00,

with a root mean squared error (RMSE) of 0.79 days. From this, it can be assumed that predicting the plant age is a relatively easier task than predicting the plant's rotation.

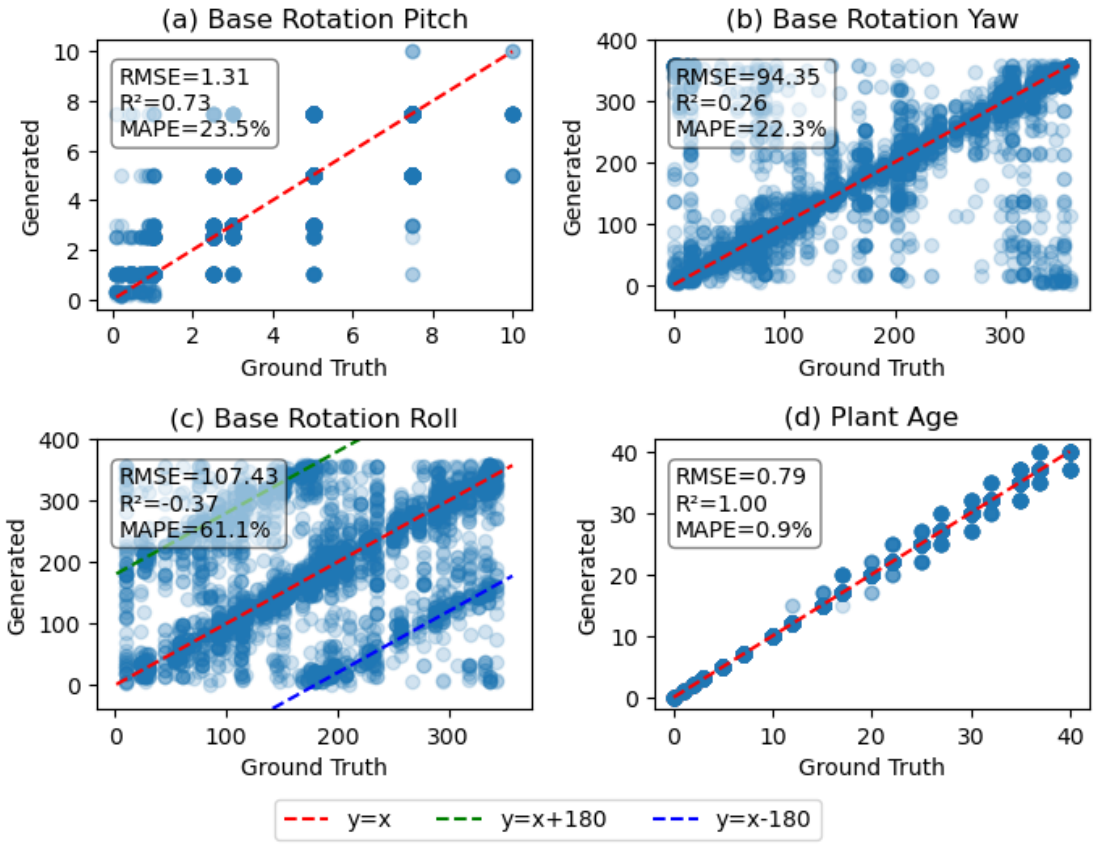


Figure 4.7. Comparison of base rotation angles (a) pitch, (b) yaw, and (c) roll of the first unifoliate shoot and (d) plant age between the generated and ground truth token sequences. The tokens for base rotation and plant age were converted to decimal values for comparison.

As mentioned previously, the number of plant organs can be an important trait for plant phenotyping. Therefore, the number of plant organs from the generated plant architecture model and the ground truth was compared by counting plant organ tokens. The number of shoots refers to the number of shoot tokens in the plant architecture token sequence, which is a signal for creating a new branch. The number of phytomers was counted by grouping tokens that appear in the order of internode, petiole, and leaf into a single phytomer.

Comparing the total number of plant organs revealed a high correlation between the gen-

erated and the ground truth plant architecture, similar to the correlation observed with plant age. Fig. 4.8 (a) shows the scatter plot of the number of shoots, with an R^2 value of 0.83 and an RMSE of 1.16. However, when compared to the 1:1 line, it was observed that the number of shoots predicted by the model was 20.7% less than the ground truth. Even if the generated plant architectures have fewer shoot tokens, as shown in Fig. 4.8 (b), the model remains accurate in terms of the total number of phytomers in the plant architecture model.

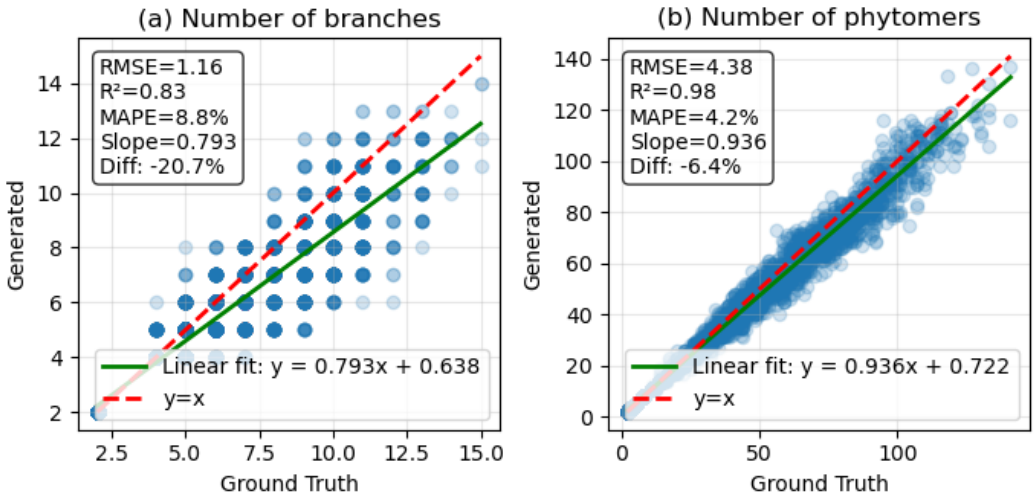


Figure 4.8. Comparison of the (a) number of branches and (b) phytomers from the generated token sequence. Ground truth refers to the true counts in the dataset and generated refers to the result obtained from the generated token sequence. Green lines represent the fitted linear regression between the ground truth and the generated data, while red dotted lines represent the y=x fit. Based on the slopes of the linear fit, the slope differences are displayed as Diff, showing –20.7% for (a) number of branches and –6.4% for (b) number of phytomers.

The distributions of plant organ parameter tokens from generated plant architecture tokens and ground truth are illustrated in Fig. 4.9. As shown in Fig. 4.7, the model could estimate primary branch rotation and plant age parameters; these estimated parameters showed good agreement with the ground truth distributions, with a normalized Wasserstein distance of less than 0.05. Parameters that determine the size of each plant organ, such as internode length & radius, petiole length & radius, and leaf scale distributions, were also found to follow the ground truth distribution closely. However, some of the pa-

rameters showed normalized Wasserstein distance higher than 0.05, such as internode phyllotactic angle (0.059) and leaf pitch (0.065).

The leaf angle distribution of the plant architecture was calculated through geometric calculations based on leaf normals generated in the plant architecture simulation. A comparison was then made between the ground truth data and the model-generated leaf angle distributions. Fig. 4.10 shows the results of calculating the leaf inclination angle using the plant architecture token sequence predicted from the image. Overall, the agreement ranged from a minimum of 0.018 to a maximum of 0.078 based on the normalized Wasserstein Distance. In the case of Fig. 4.10 (a) for day 5, the generated plant architecture predicted leaf angles in the 18° to 36° range and above 45° to 63° with a higher frequency than the ground truth, while predicting a lower frequency in the 36° to 45° and 63° to 90° ranges. For day 10 in Fig. 4.10 (b), it predicted a higher frequency in the 18° to 36° range and similar or lower frequencies in the other ranges. In the cases of Fig. 4.10 (c) - (e), although some ranges were predicted with varying accuracy, the overall trend shows a normalized Wasserstein distance within 0.1, or a 10% difference.

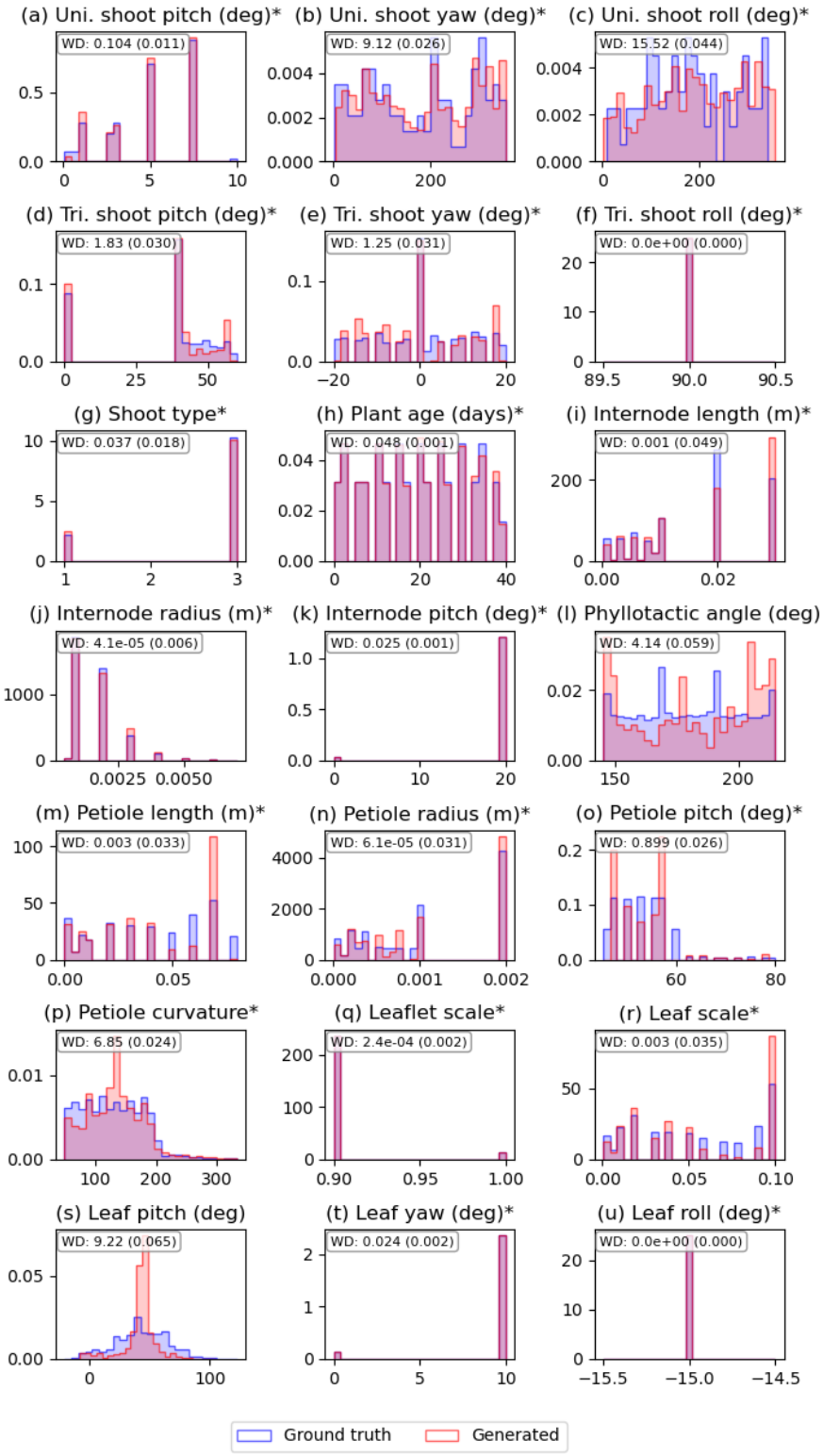


Figure 4.9. Parameter distribution comparison between generated and ground truth plant architectures from the test dataset (n=4,000). (a–c) Unifoliate (Uni.) and Tri. (trifoliate) shoots indicate branches bearing unifoliate and trifoliate leaves, respectively. Asterisks (*) denote normalized Wasserstein distances (WD) below 0.05 (5%). The x-axes represent the parameters denoted in the subplot titles and the y-axes represent probability density.

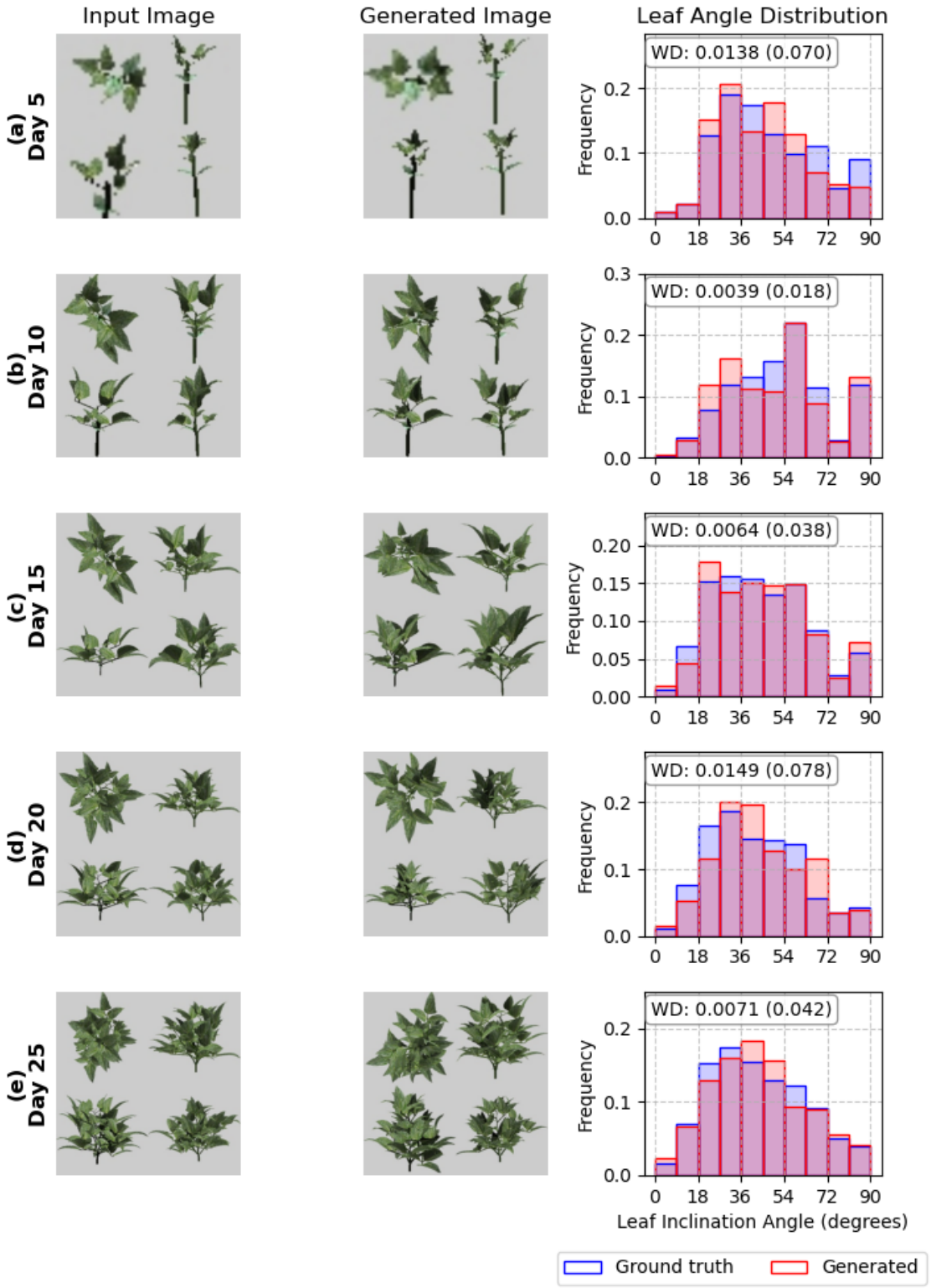


Figure 4.10. Selected sample images from a test plant across the plant age dates (a-e). Left column: ground truth images from the dataset. Middle column: Re-rendered image from generated plant architecture. Right column: Comparison of leaf angle distribution between ground truth and generated, along with Wasserstein distance.

4.3.3 Plant Bulk Trait Estimation and Comparison with Regression Baseline

This section calculates plant height, leaf count, and total leaf area using Helios plant simulation for 4,000 pairs of ground truth and generated plant architecture token sequences and compares them against a regression-based baseline method. Fig. 4.11 illustrates the results of a baseline model utilizing vision transformer features with a fully connected layer to estimate plant height, leaf count, and leaf area. As shown in Fig. 4.11, the ViT-B + FC model, which directly predicts plant traits using features from ViT-B, yields mean absolute percentage error (MAPE) values of 3.5%, 25.9%, and 11.9% for plant height, leaf count, and leaf area, respectively. Our model, which is shown as ViT-B + LM head, showed a higher MAPE value of 4.9% for estimating the plant height compared to the baseline method. However, leaf count and leaf area showed lower MAPE values than the baseline method, showing 4.1% for leaf count and 3.2% for leaf area.

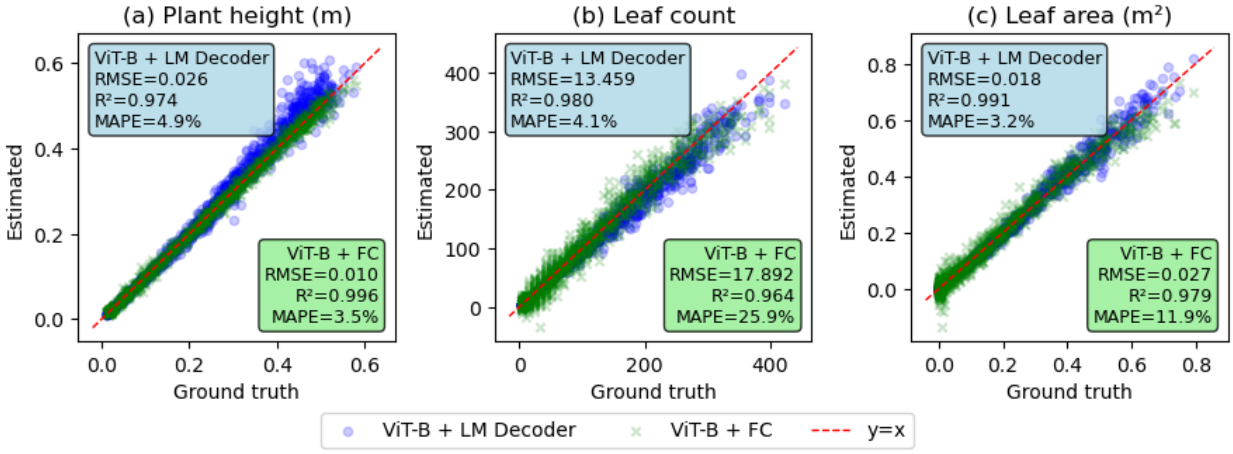


Figure 4.11. Comparison of plant scale bulk parameters: (a) Plant height, (b) leaf count, and (c) leaf area, derived from plant architecture generation and direct prediction using image features.

4.4 Discussion

4.4.1 Effect of Model Size on BLEU-4 and ROUGE-L Scores

As shown in Table 4.3, larger model configurations demonstrated better performance in the autoregressive generation process. The BLEU-4 score reached 94.00% for the best-

performing model (ViT-B with gpt-2-medium, 448-pixel multi-view input). However, this metric should be interpreted with caution: the highly structured and repetitive nature of plant architecture token sequences, combined with limited organ types and quantized parameter values, can artificially inflate BLEU scores through high n-gram overlap even when individual architectural parameters are imperfect. Nevertheless, BLEU-4 remains useful as a relative metric for comparing model configurations, as higher scores generally reflect better recovery of organ-level topology and parameter combinations. Of the generated token patterns, 94.00% matched sequences in the ground truth corpus, while the remaining 6.0% represented novel or incorrect patterns.

The size of the model and the amount of information did not always correlate with BLEU-4 scores. Notably, the smallest model configuration (ViT-S + gpt-2, 224-pixel top-view) achieved a BLEU-4 score of 92.35%, only 1.65 percentage points below the best-performing setup (94.00%). This minimal gap suggests that the constrained token vocabulary limited to 24 organ tokens and quantized parameter values may bound the task complexity, enabling smaller models to capture most structural patterns. While ROUGE-L scores showed more consistent improvements with larger image sizes, multi-view data, and increased model capacity, the small absolute differences indicate that dataset scale or token representation richness, rather than model capacity alone, may be the limiting factors for further performance gains.

4.4.2 Plant Architecture Generation Accuracy

The main hypothesis of this paper was whether the details of plant architecture and topology can be learned from images using a large language model. We tested this hypothesis by using synthetic image data alongside detailed plant architecture information and training a vision-based large language model algorithm. Research predicting plant structure through images has been ongoing; however, this is the first study to utilize a vision language model to predict tokens that define plant structure. This study demonstrated that

a vision language model can generate plant architecture, from organ-level to plant-level, from an image by estimating the tokens that build plant architecture.

As shown in Fig. 4.6, even if the generated plant architecture overall appeared to reflect the general shape and complexity of the ground truth, it could not perfectly replicate the input image itself. This is because the model is only trained to predict the n^{th} token from a sequence of tokens up to n-1 during the model training process. In other words, the model cannot access the rendered images generated from plant architecture tokens, and therefore, it has not been optimized to maximize the similarity of these images. Also, the generated plant architecture tends to have fewer branches than the ground truth, showing the model could not generate the highly nested branching structure of the plants.

4.4.3 Plant Trait Estimation Accuracy

Even though there is still room for improvement in the generated plant architecture, the results demonstrate that the model successfully estimated key plant traits from the input image. Figs. 4.7 to 4.9 showed that the plant architectural parameters can be estimated from the generated plant architecture. Still, there were some limitations in estimating the plant organ parameters. First, converting floating-point value parameters to integer token IDs introduced quantization error, transforming the parameter distributions into discrete distributions and changing the shape of the distribution. For example, the unifoliate shoot's pitch angle and internode phyllotactic angle were sampled from uniform distributions Table 4.1, but the tokenized distributions Fig. 4.9 (a) and (i) were not uniformly distributed. Secondly, a phenomenon called mean collapse occurred, where the data tends to cluster around the most frequently occurring values in the ground truth data, resulting in predictions collapsing to a few representative tokens. This is noticeable in the leaf parameters, such as in the leaf pitch values from Fig. 4.9(s), where up to 400 leaves can appear in a single plant architecture model. Still, the overall distributions of estimated plant organ parameters followed the parameter distributions of the ground truth, showing a maximum

value of normalized WD as 0.065.

By analyzing the generated plant architecture at the plant scale, indirect traits such as leaf angle distribution can be derived, as shown in Fig. 4.10, that can only be calculated from the 3D simulation and were not used for training explicitly. Although the leaf pitch angle distributions showed limited representation, the final calculated leaf angle distribution can be predicted with relatively high accuracy, showing normalized WD values of 0.018 to 0.078 Fig. 4.10. This indicates that while the model cannot perfectly predict the angle of every individual leaf, it can accurately capture the overall trend of leaf angle distribution in the image without being explicitly trained to do so.

4.4.4 Estimation of Plant Simulation-based Traits

In addition to plant organ parameters, plant-scale bulk parameters, including plant height, leaf count, and leaf area, were compared between the predicted and the ground truth. Even though the baseline model only utilized a pretrained feature extractor with one layer of a fully connected layer, it could successfully predict those bulk parameters with high R^2 values, by explicitly learning those traits from image features. However, the baseline approach is limited to predicting individual traits and cannot capture the overall complex architecture of the plant, which is needed for FSPM simulation inputs. Compared to this, our model implicitly learned plant height, leaf count, and leaf area through the plant architecture token sequence, and it was able to predict these variables more accurately than models trained directly on them. Notably, leaf count Fig. 4.11 (b) and leaf area Fig. 4.11 (c) could be predicted with higher R^2 and lower RMSE than direct estimation. The better accuracy on leaf count and leaf area means that the model, trained using plant architecture information, can more accurately predict traits that are difficult to estimate due to occlusion. In other words, the model captures patterns in the plant's internal architecture — leaf count and leaf area that are difficult to estimate from 2D images alone due to occlusion — and uses those patterns to estimate structural traits more accurately than

methods trained directly on those traits.

4.4.5 FSPM Validation

The ultimate usage of generated plant architectures is in their ability to preserve functional plant behaviors when used as inputs for plant biophysical simulations. The current evaluation focuses on NLP performance, plant architecture topology, plant organ parameters and bulk traits against ground truth. Therefore, functional validation of these generated plant models with FSPM simulations is an important next step.

Future work will therefore address this by simulating biophysical parameters using both ground truth and generated architectures in Helios, comparing outputs such as canopy photosynthesis, light interception efficiency, and energy balance calculations. This validation is critical because small architectural deviations that appear acceptable under token-based or parameter-based metrics may compound to produce significant errors in FSPM. As the Helios plant simulation can compute biophysical processes from a generated XML file, we can systematically evaluate how reconstruction errors at the organ level propagate to plant-level functional outputs. This will enable identification of which architectural parameters most critically influence functional performance, guiding targeted improvement to the tokenization and generation process. For instance, while the current model achieves reasonable accuracy in leaf angle distribution Fig. 4.10, validation against photosynthetic output would confirm whether these distributions capture functionally meaningful canopy light interception patterns.

4.4.6 Synthetic-to-Real Transfer Challenges

While this study demonstrates the feasibility of generating plant architecture from synthetic images, several factors present substantial challenges for transferring to real-world applications. The deterministic nature of synthetic rendering, uniform backgrounds, consistent illuminations, and idealized plant textures creates a domain gap between the synthetic and real domains where variable lighting, complex backgrounds, and occlusions are

common. Developmental variability in real crops, including pest damage, nutrient stress, and environmental effects may not be present in procedurally generated synthetic plants. The camera positions in the synthetic dataset are fixed and known, whereas field data collection may involve positional error and image blur. These domain gaps suggest that performance may degrade when applying the trained model to real imagery without domain adaptation techniques such as few-shot learning with field-collected data or extensive data augmentation during synthetic training. Addressing these challenges presents a critical next step for practical development.

4.5 Conclusions

This research hypothesized that organ-level plant architecture can be generated from implicit information in plant images through a vision-language model. The results showed that a vision encoder-decoder model trained on a synthetic cowpea dataset could generate organ-level plant architectural reconstructions from images. Also, the model could estimate whole-plant traits such as plant height, leaf count, and leaf area, approaching directly trained models' performance even if the language model never saw those values during training. The main takeaway of this paper is that plant architecture can be processed and tokenized for training with a large language model. The benefit of the synthetic dataset is that it provides accurate and comprehensive ground truth annotations that are practically infeasible to obtain through field measurements. The synthetic dataset's accurate and comprehensive plant architecture information allows us to verify the algorithm's validity with "exact" ground truth data. Future research will extend this study to real image domains. We will improve this result by collecting real plant architecture data from the field to compare with our model's predictions and improve performance using few-shot learning with the collected data. Also, we will test the algorithm with various backgrounds, lighting conditions, and occlusions to overcome the complexity of the actual environment.

The methodology used in this work has limitations that will be the target for improvement

in future iterations. The current model training is an open-loop process, which means the rendered image from the generated plant architecture cannot provide feedback to the model weights. Therefore, future research should provide feedback on the generated image through reinforcement learning by adding the Helios simulation program to the training loop. We plan to incorporate reinforcement learning based on rendering image similarity into this pre-trained plant architecture token generation model, aiming to fine-tune the model so that the final rendered image is similar to the input image. At present, only the plant organs of the shoot, internode, petiole, and leaf are considered, assuming the plants are in the vegetative growth stage; however, reproductive organs will be included in future studies.

In conclusion, this paper demonstrates that vision-language models can reconstruct plant architecture from images. The synthetic dataset provides exact ground truth for validation, and the method can be extended to real imagery and larger datasets in future work.

Chapter 5

Using Vision Language Foundation Models to Generate Plant Simulation Configurations via In-Context Learning

Abstract

This chapter introduces a benchmark for evaluating whether vision-language models (VLMs) can generate plant simulation configurations from imagery using in-context learning. We study this benchmark for cowpea plot reconstruction for plant simulations, where the VLM needs to generate structured JavaScript Object Notation (JSON) configurations that include field and plant information. Open-weight multimodal models from Gemma 4 and Qwen3.5 families are evaluated on a synthetic cowpea dataset with known JSON ground truth and on a real drone orthophoto dataset with field-collected JSON. Five in-context learning methods are used, from format restriction instruction to few-shot image examples with auxiliary grounding information. The results show that VLMs can generate valid JSON outputs, can generally estimate days after planting (DAP), plant counts, plant locations, sun angles, and leaf chlorophyll content, and can render approximate simulations of cowpea plots. Error metrics vary across model families and often remain worse than dataset baselines, particularly when VLMs' pretrained knowledge dominates over weak visual evidence. These results position image-to-simulation JSON generation as a promising but currently challenging task, and establish a benchmark for studying how multimodal reasoning, prompt design, and the sim-to-real domain gap affect plant phenotyping tasks.

5.1 Introduction

Plant modeling and simulation help analyze plant performance and interactions with the environment. They also help researchers understand the relationship between plant genotypes and their shape, and design ideal plant types. Detailed 3D representations of the field, plant positions, and canopy architecture are essential for simulating biophysical processes such as photosynthesis and water use (Gaillard et al., 2020; Soualiou et al., 2021).

Functional-structural plant modeling (FSPM) provides a framework to simulate plant behavior based on plant architecture, considering biophysical and environmental factors, from molecular and cellular levels, to the organ level, and up to the ecological level (Soualiou et al., 2021). For FSPMs to be useful and reproducible, they require structured representations of plot geometry, environmental conditions, and crop 3D architecture, because simulation quality depends on how accurately those inputs describe the real world.

To address this need, FSPMs often use structured languages to define plants and their parameters, such as L-systems (Lindenmayer, 1968a,b), XL (Kurth and Ong, 2017), and L-Py (Boudon et al., 2012). Most previous studies used a forward workflow where experts define rules and structures, from which simulations or images are then generated (Baker et al., 2024). However, to faithfully represent real-world plants, an inverse approach is needed: generating plant models from sensing data. Specifically, research on generating FSPMs from images for multiple plants at the plot or field scale is still lacking.

To bridge the gap between field-scale sensing data and plant modeling, recent studies have focused on extracting architectural measurements from sensor data. For example, Zhu et al. (2023) reconstructed organ-scale phenotypic parameters including leaf length, width, and area from 3D point clouds of field crops using unmanned aerial vehicle (UAV) imagery. Similarly, Chebrolu et al. (2021) developed methods for registering spatio-temporal point clouds to track plant traits at the organ level over time, while Ziamtsov and Navlakha

(2019) used deep learning to classify plant organs and skeletonize stems from 3D point clouds with high accuracy. However, these extracted measurements still need to be transferred into structured FSPM definitions to enable functional simulations.

While organ-level plant architecture generation from single images has demonstrated that structured plant representations can be recovered from visual inputs (Chapter 4), scaling this capability to plot-level simulation configurations that encode multiple plants, environmental conditions, and biophysical parameters remains untested. Therefore, this study extends the image-to-simulation pipeline from single-plant XML generation to field-scale JSON configuration generation.

Recent large language models (LLMs) and vision language models (VLMs) have shown that models can generate structured document outputs such as JSON, HTML, and tabular representations from textual or visual context (Lee et al., 2022; Zhang et al., 2023). These advancements have raised the possibility that a multimodal model could directly infer plant simulation configuration documents from plot images, rather than relying on manual parameterization pipelines. However, the feasibility of this idea depends not only on generating valid structured outputs from image context, but also on recovering biologically accurate parameters from the image.

Agricultural VLM benchmarks have expanded rapidly to evaluate the feasibility of VLMs for agricultural perception tasks (Arshad et al., 2025; Awais et al., 2025; Shinoda et al., 2025; Yang et al., 2025; Zhou et al., 2024). Existing agricultural VLM benchmarks mainly focus on classification, identification, question answering, and related agronomic perception tasks. While these benchmarks have broadened the evaluation of VLMs in agriculture, there is still no benchmark that tests whether VLMs can generate simulation-ready plant configuration files that combine multiple tasks, such as plant localization and the estimation of environmental and biophysical traits within a structured document format. It remains unclear whether current VLMs can support image-to-simulation workflows.

To address this gap, we make three main contributions:

- A benchmark for image-to-plant-simulation JSON generation, comprising synthetic cowpea plots with known ground truth and a real drone orthophoto dataset with field-collected ground truth.[1]
- A systematic evaluation of six open-source VLMs under five in-context learning methods, quantifying JSON integrity, geometric accuracy, and biophysical parameter estimation.
- An analysis of how fine-tuning, few-shot data modality, and contextual bias affect VLM performance on plant simulation generation.

The remainder of this chapter is organized as follows. Section 5.2 describes the cowpea dataset, simulation pipeline, JSON schema, and the VLM evaluation protocol. Section 5.3 presents the results across model families and in-context learning methods. Section 5.4 discusses the implications of the results. Section 5.5 concludes with limitations and future directions.

[1]Dataset is publicly available here: https://huggingface.co/datasets/heesup/vlm-plant-sim

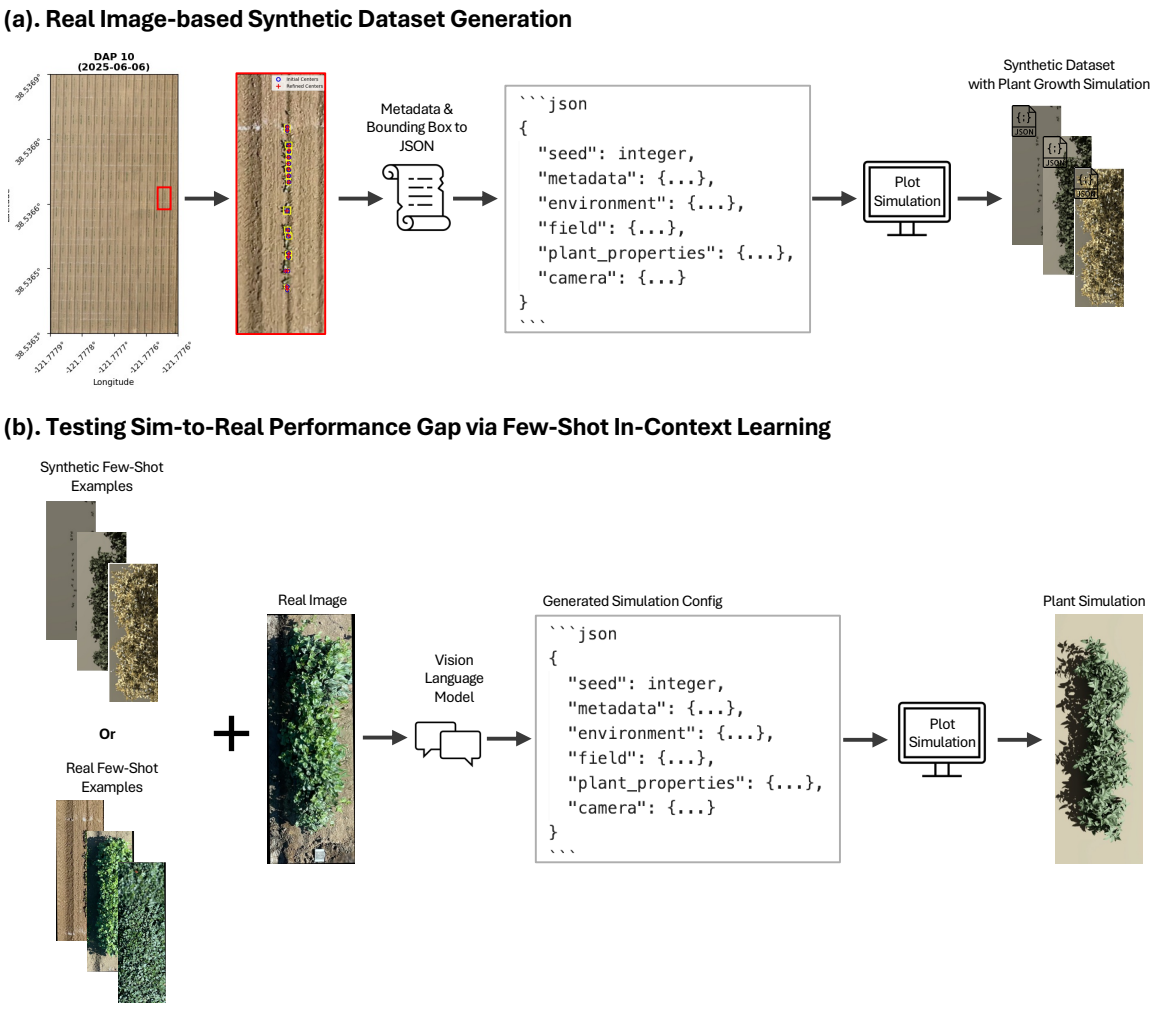


Figure 5.1. Overall diagram of generating datasets and evaluating vision-language models for plant simulation generation. (a) Synthetic Dataset Construction: Real drone orthophotos undergo plant detection and spatial feature extraction to generate structured JSON configurations, which drive procedural plot simulations across growth stages (10–90 DAP). (b) Sim-to-Real Evaluation: Vision-language models receive few-shot examples (synthetic or real) paired with a target real image to generate simulation configurations via in-context learning, enabling cross-domain performance assessment.

5.2 Methods

5.2.1 Drone Remote Sensing Dataset

Drone-based remote sensing of a cowpea (Vigna unguiculata (L.) Walp.) field was conducted in 2025 at a breeding trial field in California. The plot dimensions were 1.5 m by 3.0 m with 1.5 m alleys separating each plot, and the plants were planted in a single row within the plot. The field had 40 beds and 14 tiers with borders to minimize edge effects.

60 cowpea genotypes were planted from beds 2 to 16 with three replicate blocks, and the rest of the field was planted with common bean and lima bean populations. The cowpeas were planted on May 27, 2025, and harvested on October 9, 2025.

Drone imagery was collected in 2025 using a custom ArduPilot-based platform equipped with an iPhone 13 Pro mounted on 3D-printed attachments, flown at 10 m above ground level. The 20-minute flight missions were conducted on 10, 28, 49, 70, and 87 days after planting (DAP), with 80% side overlap and 70% frontal overlap. The orthophotos were processed with OpenDroneMap and exported as GeoTIFF files. Plot boundaries were annotated using QGIS, and the plot images were cropped accordingly.

Ground truth DAP values were calculated as the difference between the drone data collection date and the planting date. Plant counts and locations were manually annotated at 10 DAP images using LabelMe software, saved in common objects in context (COCO) JSON format. These annotations were converted to plant center coordinates (in meters), and used as ground truth for the rest of the season. Because individual plants remained rooted in fixed positions after emergence, as shown in Fig. 5.2, we used these center locations as reference geometry across later dates.

Sun elevation and azimuth angles at the time of image capture were calculated in degrees using the pvlib Python library (Anderson et al., 2023), based on the plot center's latitude, longitude, and the exact image capture timestamp. Ground truth chlorophyll values were sampled three times per season in 24 of the 180 cowpea plots using a chlorophyll content index (CCI) meter (Apogee MC100, Apogee Instruments, Inc., USA) and converted from CCI units to chlorophyll content $\mu g/cm^2$ using an empirical equation derived from a previous study (Parry et al., 2014). The measured time series of chlorophyll content per plot was fitted to polynomial functions of DAP and resampled for the drone data collection dates. The DAP, plant count, plant locations, sun elevation, sun azimuth, and chlorophyll were used as the benchmark dataset for further evaluation.

Figure 5.2. Temporal progression of a representative cowpea plot (bed 2, tier 2) across the growing season. Images show canopy development from early emergence (10 DAP) to maturity (87 DAP), with red bounding boxes indicating initial plant locations tracked throughout the season. Days after planting (DAP), sun elevation and azimuth angles (El/Az) were calculated from image metadata. Plant count and locations were annotated from the DAP 10 image. Leaf chlorophyll content (Chl) was derived from polynomial fittings of the field measurement.

5.2.2 Cowpea Plot Simulation JSON

A structured JSON schema was used to capture the minimum information required to reconstruct a cowpea plot in the simulation environment. The schema includes metadata, environment, field layout, plant properties, and camera parameters. Some fields are visually inferable from imagery, such as DAP, whereas others are metadata, such as year and location. The JSON file has six top-level fields: random seed, metadata, environment, field, plant properties, and camera setup.

The JSON file starts with a random seed for reproducibility, which controls the random sampling used to procedurally generate plants and sample the parameters in the simulation. The metadata key includes year, location, plant type, and days after planting. The environment key contains soil and sun environment data, such as soil spectral data cat-

egories and specular reflection coefficients, as well as sun elevation and azimuth degrees. The field key specifies the field layout, including plot size and the number of beds. Plot keys are nested under the field key, and the plot key contains plot bed and row ID, and individual plant locations as an array. The plant properties key includes a chlorophyll parameter that determines the leaf color spectrum. Image sensor and camera positioning related parameters are in the camera key, such as shutter speed, ISO, camera resolution and model, camera height, and a look-at vector.

5.2.3 Cowpea Plot Simulator

A plant simulator program was developed using the Helios 3D simulation library (Bailey, 2019) to generate synthetic cowpea plot images from plant simulation JSON files. The program generates a synthetic cowpea plot with full 3D plant geometry by reading initial plant locations, DAP, solar position, chlorophyll content, and camera parameters. The Helios library initializes plants on the specified plant locations, and advances time forward to grow the plant architecture procedurally up to the desired DAP based on phenological stages. The plant architectural growth is simulated with phyllochron control, which is the minimum number of days between successive phytomer emergence along a shoot (Bailey, 2025). All the growth and shape parameters were inherited from default plant architecture library settings.

To colorize the generated 3D plant architecture model, leaf and stem spectra were generated from the PROSPECT leaf optics model (Jacquemoud and Baret, 1990), while flower and pod spectra were generated from a mixture of standard purple and yellow spectra. To simulate various lighting conditions, sun elevation and azimuth were randomly sampled from uniform distributions over [0°, 90°] and [0°, 360°], respectively.

5.2.4 Synthetic Cowpea Dataset

To generate synthetic data following real-world field configurations in a scalable way, an automated pipeline was developed. Fig. 5.1 (a) shows the overview of the synthetic image

generation process. The Excess Green (ExG) (D. M. Woebbecke et al., 1995) vegetation index was applied to segment plants from the background, and plant blobs were detected using OpenCV blob detection functions. The detected blobs were divided into individual plants based on the blobs' aspect ratio, and the center positions were refined to the center of gravity of each blob. The plant locations were calculated considering the pixel locations and the plot dimensions in meters, with the image center set to the origin (0,0).

The plant stand count and plant locations were detected from the drone orthophoto from the 10 DAP across 560 plots, and the detected plant locations were saved in JSON format to simulate the cowpea plants in the plant simulator. Starting from 0 DAP, cowpea plant growth simulations were performed for 10, 30, 50, 70, and 90 DAP, resulting in a total of 2,800 synthetic cowpea plot simulations. The simulated plots were exported to JSON ground truth files and corresponding JPEG images in 381 $\times$ 1080 resolution.

5.2.5 Vision-Language Models

To evaluate the VLMs' ability to generate a simulation config JSON file from an image, two families of vision-language models were tested. We selected Gemma 4 (DeepMind, 2026) and Qwen3.5 (QwenTeam, 2026) because they represent recent open-source multimodal model families with various model sizes, strong instruction-following and reasoning abilities, and practical support for controlled local inference without API costs. Gemma 4 was released in April 2026 and is designed to run on edge devices, with model sizes ranging from 2B to 31B parameters. Qwen3.5 was released in February 2026 and includes models from 0.8B to 122B parameters. We tested the 2B, 4B, and 31B Gemma 4 models and the 4B, 9B, and 27B Qwen3.5 models. A self-hosted vLLM (Kwon et al., 2023) server was used to provide an API endpoint for interacting with the models via a Python script. The maximum context size was set to 32K to accommodate the maximum length of in-context learning methods and generation results, and the temperature was set to 0 for deterministic generation.

5.2.6 In-context Learning

LLMs can learn from context and provide an answer based on the given information without updating the model weights, a capability called few-shot learning or in-context learning (Brown et al., 2020). We tested in-context learning methods by progressively providing context to the model and observing changes in JSON generation performance. A summary and examples of the five in-context learning methods are shown in Table 5.1.

The simplest approach is to provide format restriction instructions (FRI), such as "Respond as JSON" or "Provide your output in the following valid JSON format". Therefore, the first method (M1) served as a baseline that defined the LLMs' role, task, goal, and JSON FRI. The second method (M2) added a JSON schema to the first, so the models could reference the JSON structure and variable types. The third method (M3) added a few-shot JSON examples to the second method. Three JSON files were selected for 10, 50, and 90 DAP for the synthetic dataset, and 10, 49, and 87 DAP from the real dataset. For each selected DAP, samples that maximized the variance in plant count, sun elevation, sun azimuth, and chlorophyll were selected.

An important element from the example JSON was the reasoning key text, which provided references for building step-by-step reasoning text to encourage intermediate decomposition of the tasks. Following this, VLMs were instructed to include the 'reasoning' field before producing the structured output. This two-stage process enables the model to reason freely in natural language, then complete the remaining JSON structure based on the analysis (Tam et al., 2024). The reasoning field was treated as a prompting aid to help the model rather than as a validated explanation of the models' reasoning and was not included in the evaluation metrics.

The fourth method (M4) included few-shot image-JSON pairs, so that the models could learn how to generate JSON from images by extracting visual context. Three pairs of images and JSON were delivered as a list of messages between the user and the assistant,

Table 5.1. In-context learning configurations and example texts added to the prompt. "... (more)" signifies abbreviated content for presentation purposes and was not part of the prompt input.

Configuration	Example	
Method 1: Zero-shot JSON generation	You are a plant phenotyping expert analyzing Helios 3D plant simulator images. Extract all simulation parameters that produced this image and output them as JSON. — Parameter Reference — 1. metadata ... (more) Answer:	
Method 2: Method 1 + **JSON schema**	`JSON SCHEMA: { "reasoning": "string", "seed": "integer", "metadata": { "year": "integer", "location": "string", "plant_type": "string", "DAP": "integer" }, ...(more) }`	
Method 3: Method 2 + **few-shot JSONs**	Example 1: `{ "reasoning": "Visual analysis: Plant maturity suggests 10 days growth. ...(more) }`	×3
Method 4: Method 3 + **few-shot images**	Example 1: user: < $IMAGE$ > A: `{ "reasoning": "Visual analysis: Plant maturity suggests 10 days growth. ...(more) }`	×3
Method 5: Method 4 + **grounding info**	Ground truth hints for target image: Plant age: 10 DAP, Plant count: 14, Sun position: 62.9° elev., 169.4° azim., Plant locations (rx, ry): [(0.163, 0.073), (0.162, 0.085), ..., (0.174, 0.743)]. Convert to meters: $x = (r_x - 0.5) \times 1.3521$, $y = -(r_y - 0.5) \times 3.8405$	

simulating three pairs of questions and answers.

The fifth method (M5) provided the model with auxiliary grounding hints. M5 added exact DAP, solar positions, and relative plant locations to M4, but absolute plant locations and chlorophyll content were not included. We designed M5 to read DAP, sun elevation, and azimuth from the grounding text, convert relative plant locations to absolute locations, and estimate chlorophyll content from the image without any hint. This method tests whether VLMs can reliably integrate structured hints in plain text into a valid JSON simulation configuration, and it assesses the language performance of the models and how chlorophyll content estimation accuracy changes if the rest of the parameters were given.

5.2.7 Evaluation Metrics

The JSON output quality was evaluated in three categories: JSON integrity, geometric accuracy, and lighting/biophysical accuracy. All six models were evaluated on a synthetic cowpea dataset, and the few-shot examples provided to the models were excluded from the evaluation.

JSON integrity: We evaluated JSON integrity metrics to assess quality differences in JSON generation across models and contexts, focusing on structural reliability of generated JSON files. The first metric was the JSON syntax error rate, the percentage of the response containing JSON syntax errors that could not be parsed into JSON,

$$\text{Syntax Error Rate (\%)} = \left(1 - \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\text{valid}}(x_i)\right) \times 100 \ , \tag{5.1}$$

here N is the total number of generated responses, x_i is the i-th generated response, and $\mathbb{1}_{\text{valid}}(x_i)$ is an indicator function that equals 1 if x_i can be successfully parsed into a valid JSON dictionary, and 0 otherwise.

The second metric was the JSON key-missing rate, which counts the number of missing keys in the response and divides by the total number of JSON keys in the ground truth,

$$\text{Key Missing Rate (\%)} = \frac{1}{|V|}\sum_{v \in V}\left(1 - \frac{K^{(v)}_{\text{present}}}{K^{(v)}_{\text{expected}}}\right) \times 100, \tag{5.2}$$

where V is the set of all successfully parsed JSON responses (where $\mathbb{1}_{\text{valid}}(x_i) = 1$),

$K^{(v)}_{\text{expected}}$ is the total number of expected keys (including nested properties) defined in the ground truth schema for response v, and $K^{(v)}_{\text{present}}$ is the number of those expected keys that are successfully found in the generated response v.

Geometric evaluations: The geometric evaluation metrics represent plant geometric er-

rors indirectly, by comparing the DAP, plant count, and plant locations. Because DAP caused major geometric changes in a cowpea plot by altering plant size, architecture, and plant organ visibility, as shown in Fig. 5.2, DAP was categorized in the geometric evaluations category. The DAP and plant count errors were evaluated using mean absolute error (MAE).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i|$$

For plant locations, spatial alignment between the predicted S_1 and ground truth S_2 plant locations was calculated using Chamfer distance (CD) (Barrow et al., 1977),

$$d_{CD} = d\left(S_1, S_2\right) + d\left(S_2, S_1\right), \tag{5.3}$$

where d(P, Q) denotes the average nearest-neighbor distance from set P to Q.

Lighting and biophysical evaluations: Sun elevation, azimuth, and chlorophyll content change the lighting and color of the cowpea plots. Sun elevation and azimuth affect shadow length, shadow direction, and canopy illumination, whereas chlorophyll content changes leaf color and apparent greenness. For sun elevation, sun azimuth, and chlorophyll content, MAE was calculated.

5.2.8 Statistical Analysis and Dataset Baselines

The effects of model family, model size, and in-context learning method were analyzed statistically. Bar graphs with 95% confidence intervals (CIs) and statistical groups were plotted for JSON syntax error rate, key-missing rate, and parameter MAEs. The CIs for JSON syntax error and key-missing rate were computed using a Bernoulli distribution, while CIs for continuous metrics were computed using a normal distribution.

Statistical differences between models within the same in-context learning setting were assessed using the Kruskal–Wallis H-test, followed by pairwise Mann–Whitney U tests with

Bonferroni correction. Statistical groups were displayed when there were significant differences between the groups. For the largest model from Gemma 4 and Qwen3.5 model families, separate bar graphs were plotted grouped by DAP, with CIs and statistical groups indicated.

median guess baseline (B_{med}) and random guess baseline (B_{rnd}) were calculated as dataset baselines. B_{med} represents performance when estimating values using only marginal distributions of parameters without image understanding. It was obtained by predicting each parameter using the median of its valid range and computing MAE against the ground truth. B_{rnd} represents randomized guessing. To reduce stochastic variability, B_{rnd} was estimated via Monte Carlo simulation by sampling 1,000 values from each parameter's distribution and calculating the MAE relative to ground truth.

For plant locations, modified versions of B_{med} and B_{rnd} were used because MAE is inappropriate for spatial coordinates. First, the median plant count was determined from the ground truth distribution and used as the number of clusters. All plant locations were pooled and K-means clustering was used to partition them into clusters. B_{med} was then calculated as the CD between the resulting cluster centers and dataset plant locations, averaged across samples. For B_{rnd}, plant x and y coordinates were randomly sampled from plant location ranges, CDs were calculated against ground truth, and the results were averaged.

5.2.9 Vision Foundation Model Baselines

To compare few-shot learning capability between VLM and vision foundation model (VFM), vision foundation model (VFM) baselines were calculated using linear models fitted to VFM embeddings. For the VFM baseline, a linear model using zero-shot VFM embeddings was fitted to three few-shot images, yielding a minimum-norm solution. A VFM, DINOv3 (Siméoni et al., 2025) trained on a satellite dataset was used, providing 1024 embedding dimensions. Let $X \in \mathbb{R}^{N \times D}$ denote the embedding matrix extracted from the

VFM, where $N = 3$ and $D = 1024$. An augmented feature matrix $\widetilde{X} = [X,\ \mathbf{1}_N] \in \mathbb{R}^{N\times(D+1)}$ was defined, where $\mathbf{1}_N$ is a column vector of ones. The linear mapping to the target plant simulation parameters $Y \in \mathbb{R}^{N\times K}$ is given by $Y = \widetilde{X}W$. $W \in \mathbb{R}^{(D+1)\times K}$ was the augmented weight matrix combining both the projection weight and the bias, and $K = 5$ for DAP, plant count, sun elevation, sun azimuth, and chlorophyll content.

Given that the number of few-shot samples $N = 3$ was smaller than the feature dimension $D = 1024$ ($N \ll D + 1$), the linear system was highly underdetermined, yielding an infinite number of exact solutions that satisfy zero fitting error. To prevent overfitting and ensure a unique and stable solution, we constrained the hypothesis space, seeking the minimum-norm solution by $\min_W \|W\|_F^2$ subject to $Y = \widetilde{X}W$ where $\| \bullet \|_F$ denotes the Frobenius norm. The optimal closed-form solution to this constrained optimization was obtained by using the Moore–Penrose pseudo-inverse $\widetilde{X}^+$ (Golan, 2012), $W = \widetilde{X}^+Y$. Since the few-shot samples were sampled independently, rows of $\widetilde{X}$ were linearly independent, so the pseudo-inverse can be explicitly computed as $\widetilde{X}^+ = \widetilde{X}^T\left(\widetilde{X}\widetilde{X}^T\right)^{-1}$. Therefore, the weights matrix was computed by $W = \widetilde{X}^T\left(\widetilde{X}\widetilde{X}^T\right)^{-1}Y$.

5.2.10 Effect of Dataset Modalities on Fine-Tuning and Few-Shot In-Context Learning

To evaluate the effect of fine-tuning and few-shot modalities on parameter estimation, we calculated Pearson's correlation coefficient (r), MAE, and symmetric mean absolute percentage error (sMAPE),

$$\text{sMAPE (\%)} = \frac{1}{N}\sum_{i=1}^{N}\frac{200\cdot|y_i - \widehat{y}_i|}{|y_i| + |\widehat{y}_i|} \tag{5.4}$$

where N is the total number of valid samples, y_i is the ground truth parameter value for the i-th sample, and $\widehat{y}_i$ is the predicted parameter value for the i-th sample.

Effect of Synthetic Fine-Tuning on Synthetic and Real Datasets Evaluation:

To test whether fine-tuning on synthetic data could improve structured generation beyond in-context learning, the largest model from each family was fine-tuned on synthetic image-JSON pairs. Therefore, we focused on the largest models because they showed the strongest in-context learning performance and were the most plausible candidates for practical application. To fine-tune the model on a small GPU cluster, parameter-efficient fine-tuning (PEFT) was performed using low-rank adaptation (LoRA) (Hu et al., 2021) with r=16 and α=16. The Qwen3.5 27B and Gemma 4 31B models were fine-tuned with approximately 124M and 133M trainable parameters, respectively. Data from beds 9 to 40 of the synthetic cowpea dataset were split 80/20 for training and validation, excluding beds 1 to 8, which were reserved for the benchmark dataset. We used the Unsloth (Han et al., 2023) Python library to accelerate training and reduce memory overhead. Each model was trained for 5 epochs with an effective batch size of 32 across 4 NVIDIA H100 GPUs for about 10 hours per model.

Effect of Synthetic and Real Few-Shot Contexts on Real Dataset Evaluation: Another scenario involved evaluating the base models with different few-shot example modalities. The key question was whether real few-shot context improves inference on the real dataset more than synthetic few-shot context. To test this, the largest model per model family was evaluated on the real dataset under two few-shot context modalities, synthetic few-shot examples and real few-shot examples, as shown in Fig. 5.1 (b).

5.3 Results

5.3.1 Synthetic Dataset Evaluation Result

Fig. 5.3 shows the evaluation results across two model families (Gemma 4 and Qwen3.5), three model sizes (2B, 4B, and 31B for Gemma 4; 4B, 9B, and 27B for Qwen3.5), and five in-context learning methods (M1–M5). Across the synthetic benchmark, the models were generally capable of producing near-complete JSON outputs, especially after a schema was

provided. However, this structural reliability did not ensure accurate parameter estimation. Performance depended strongly on model family, model size, and in-context learning methods. Several combinations of these factors remained worse than simple prior-based baselines for plant count, plant location, and biophysical variables.

JSON Integrity Evaluations: The Gemma 4 2B model showed a 2.68% syntax error rate for M2, while the Qwen3.5 4B model showed the highest error at 3.76% for M3. Adding few-shot JSON examples increased the JSON syntax error rate for the Qwen3.5 4B model. The Gemma 4 2B model with M1 showed 1.13% key-missing rate, while the Qwen3.5 9B showed 0.12% on the same method. For all models, adding a JSON schema (M2–M4) reduced the key-missing rate to 0.0%.

Geometric Evaluations: The Gemma 4 models generally showed decreasing DAP MAEs as model size increased. The Gemma 4 models showed lower DAP MAE values for M1, M2, and M4 than Qwen3.5 for all model sizes. The Qwen3.5 9B model showed the highest DAP MAE ranging from 34.29 to 38.68 days across M1–M5. In M5, grounding information reduced all models' MAE to near zero, except for the Gemma 4 31B model, which showed 1.47 days. The Qwen3.5 4B and 9B models showed higher MAE values than B_{med}, and the Gemma 4 2B model showed DAP MAE values close to B_{med}.

Plant count MAE values showed interactions between model size and in-context learning methods. All models with M1–M4 showed higher MAE values than B_{med}. Also, only Gemma 4 31B and Qwen3.5 27B models showed lower MAE than B_{rnd} for M1 and M2. Larger Gemma 4 models showed lower plant count MAE values than smaller models, but adding few-shot examples increased the MAE values for M3 and M4 unexpectedly. Models achieved a plant-count error of less than 0.97 plants using the grounding hint in M5.

The plant location CD showed that the Gemma 4 2B and 4B models performed no better than B_{rnd} for all in-context learning methods. The Qwen3.5 4B model and 9B model showed higher error than B_{rnd} for M1 and M2, and the Qwen3.5 9B model showed lower

error than B_{rnd} for M3, M4, and M5. The Gemma 4 31B model showed the lowest error for M1, M2, M3, and M4, except for M5 where the Qwen3.5 27B performed better on plant location estimation.

Notably, adding few-shot JSON or image examples often failed to improve geometric accuracy. Our results demonstrate that additional context could reinforce prompt priors more strongly than it improved visual extraction. Furthermore, even if relative plant locations were given at M5, lightweight models could not eliminate the error completely because of arithmetic errors or failure to extract information from the context.

Lighting and Biophysical Evaluations: As shown in Fig. 5.3 (f), the Gemma 4 models had lower sun elevation MAE than the Qwen3.5 models from M1 to M4. The Gemma 4 4B and 31B models showed lower MAE values than B_{med} across all methods. In contrast, small models in Qwen3.5, such as 4B and 9B models, showed higher MAE values than B_{med} or B_{rnd}, and only the 27B model showed an MAE lower than B_{med} in M4 and M5. Providing sun elevation in the context eliminated the errors, except for the Gemma 4 2B model with M5. The sun azimuth MAE results showed that the Gemma 4 and Qwen3.5 models had similar MAE from M1 to M4. Sun azimuth MAE decreased as Gemma 4 model size increased, and the largest Qwen3.5 model showed lower error than smaller models. Similar to sun elevation, adding a grounding hint to the prompt in M5 decreased the MAE below 17.96°. However, adding few-shot JSON examples or JSON-image pairs did not substantially decrease sun elevation and azimuth MAE.

Chlorophyll MAE results showed that small models in Gemma 4 and Qwen3.5 could not estimate values below B_{med}. Gemma 4 models showed decreasing MAE when model size increased, while the Qwen3.5 9B model showed higher MAE than the 4B model in M3 and M4. Adding few-shot JSON and image decreased chlorophyll MAE for the Qwen3.5 27B model in M3 and M4, and the Gemma 4 31B model showed similar MAE for all in-context learning methods. Since chlorophyll content was not included in M5, the grounding hint

did not substantially decrease MAE except for the Gemma 4 2B model.

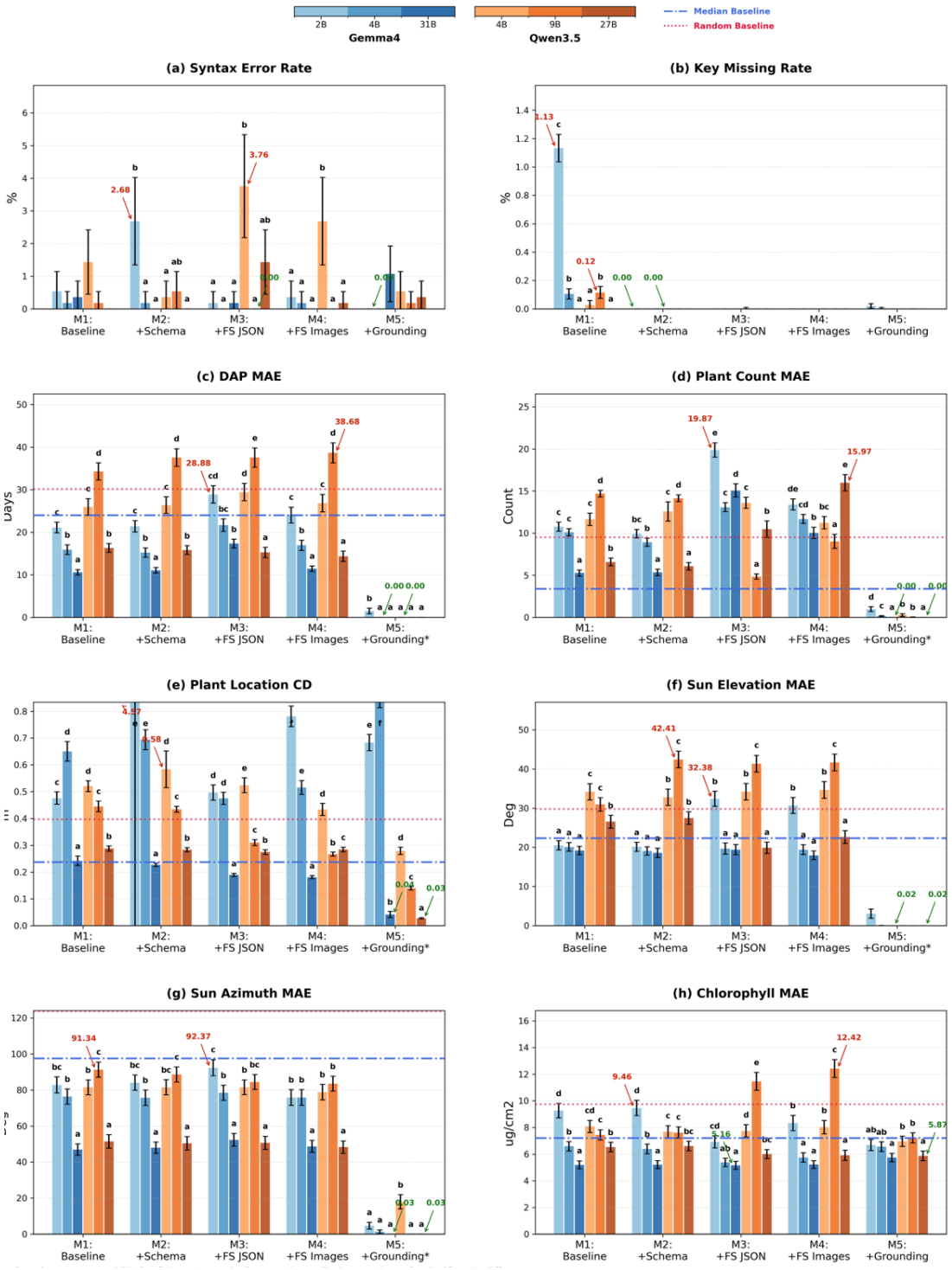


Figure 5.3. Performance of Gemma 4 (blue) and Qwen3.5 (orange) model families on a synthetic dataset across five progressive in-context learning conditions (M1–M5). Panels (a–h) show syntax error rate, key-missing rate, DAP MAE, plant count MAE, plant location CD, sun elevation MAE, sun azimuth MAE, and chlorophyll MAE, respectively. Asterisks (*) denote exact values were included in M5 ground truth hint. Dash-dotted and dotted lines indicate median and random guess baselines, respectively. Unlabeled bars or empty spaces denote an error value of exactly 0. Error bars represent 95% confidence intervals; letters indicate significant differences ($p < 0.05$).

Effect of Dataset DAP: Fig. 5.4 and Fig. 5.5 show the synthetic data evaluation result of the largest model per model family, grouped by dataset DAP. Based on Fig. 5.4 (a) and Fig. 5.5 (a), the Qwen3.5 27B model showed a maximum JSON syntax error rate of 3.6%, while Gemma 4 31B model showed maximum 1.8%. Both models showed 0.0% key-missing errors across all methods.

As shown in Fig. 5.4 (c) and Fig. 5.5 (c), DAP MAE showed a pattern increasing from DAP 10 to 30, decreasing from 30 to 70, and increasing again from 70 to 90 for both models. For 30 DAP with M1, M2, and M3, the Qwen3.5 model showed errors close to B_{rnd}, and the Gemma 4 model showed errors lower than B_{med}, except for 30 DAP with M3. The MAE for plant count increased for both models as DAP increased, especially for M3 and M4. The Qwen3.5 model showed errors higher than B_{rnd} when DAP was larger than 30 with M3 and M4.

Plant location CD increased as DAP progressed, and adding few-shot JSON and images decreased the plant location error of the Gemma 4 31B model across all DAP, while adding few-shot examples decreased plant location error at 10 and 30 DAP for the Qwen3.5 model. Sun elevation and chlorophyll MAE values increased as the DAP progressed, but sun azimuth MAE values remained similar across DAP.

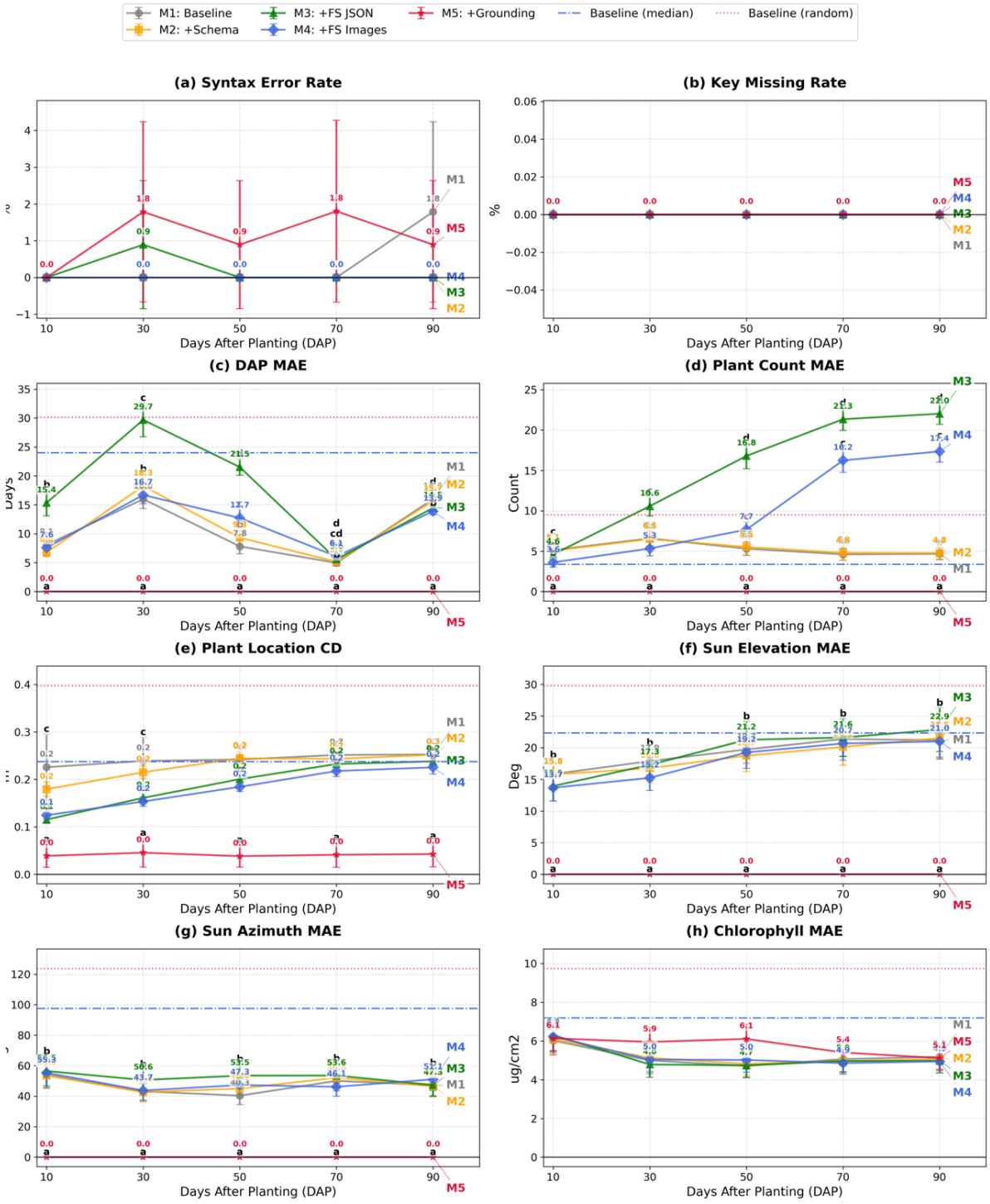


Figure 5.4. Temporal dynamics of Gemma 4 31B across growth stages (10–90 DAP) on a synthetic dataset across five progressive in-context learning conditions (M1–M5). Panels (a–h) show syntax error rate, key-missing rate, DAP MAE, plant count MAE, plant location CD, sun elevation MAE, sun azimuth MAE, and chlorophyll MAE, respectively. Asterisks (*) denote exact values were included in M5 ground truth hint. Dash-dotted and dotted lines indicate median and random guess baselines, respectively. Error bars represent 95% confidence intervals; letters indicate significant differences ($p < 0.05$) among methods at each DAP.

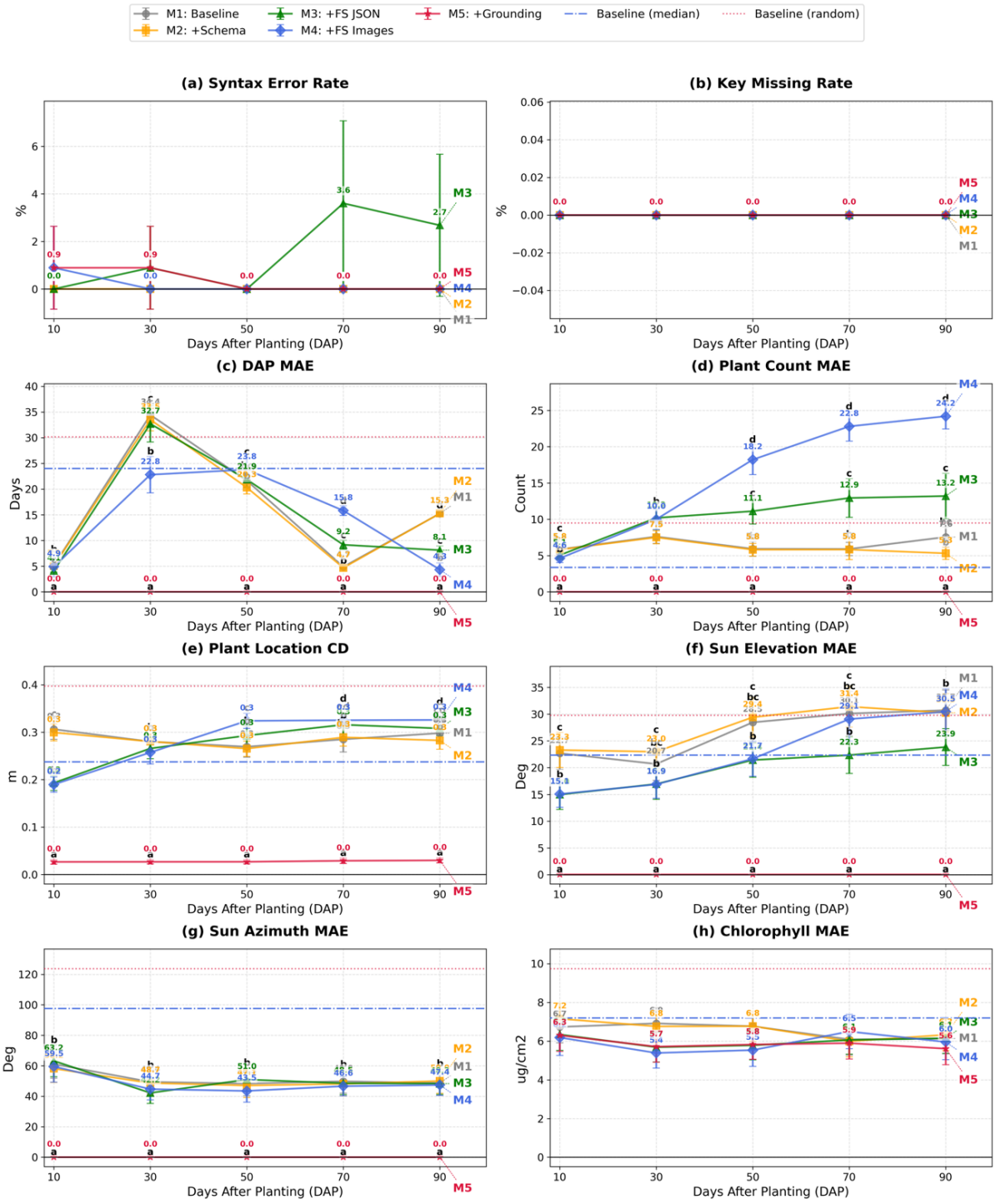


Figure 5.5. Temporal dynamics of Qwen3.5 27B across growth stages (10–90 DAP) on a synthetic dataset across five progressive in-context learning conditions (M1–M5). Panels (a–h) show syntax error rate, key-missing rate, DAP MAE, plant count MAE, plant location CD, sun elevation MAE, sun azimuth MAE, and chlorophyll MAE, respectively. Asterisks (*) denote exact values were included in M5 ground truth hint. Dash-dotted and dotted lines indicate median and random guess baselines, respectively. Error bars represent 95% confidence intervals; letters indicate significant differences ($p < 0.05$) among methods at each DAP.

5.3.2 Fine-tuned model evaluation for synthetic and real datasets

The effect of synthetic dataset supervision was tested by fine-tuning the largest Gemma 4 and Qwen3.5 models on synthetic data and evaluating them on synthetic and real datasets. Table 5.2 shows the evaluation metrics for DAP, plant count, plant location, sun elevation, sun azimuth, and chlorophyll of the fine-tuned models. During evaluations, few-shot examples were sampled from the same domain as the evaluation set.

The results show that fine-tuning on the synthetic dataset generally increased correlation coefficients and reduced MAE and sMAPE for the Gemma 4 31B model. For example, fine-tuning on the synthetic dataset increased the correlation coefficient for plant count, solar positions, and chlorophyll, while decreasing MAE and sMAPE for DAP and chlorophyll. Although the Gemma 4 31B improved modestly on the synthetic evaluation, these gains did not transfer to the real dataset, showing lower correlation coefficients and higher MAE and sMAPE values than the original model on real domain evaluation.

Fine-tuning the Qwen3.5 27B model showed degraded performance on both synthetic and real evaluations, indicating that supervised fine-tuning on the synthetic dataset may overfit the model to the training dataset and interfere with the model's pre-existing multimodal reasoning capabilities.

5.3.3 Effect of Few-Shot Modality on Real Dataset Evaluation

Table 5.3 compares the correlation coefficient, MAE, and sMAPE for DAP, plant count, sun elevation, sun azimuth, and chlorophyll between zero-shot, synthetic few-shot (Syn FS), and real few-shot (Real FS) settings. As shown in Table 5.3, providing synthetic few-shot examples to the Gemma 4 31B model improved the average correlation coefficient from 0.48 to 0.53 but increased average sMAPE from 32.4% to 45.4%, compared to the zero-shot setting (M2). Similarly, the Qwen3.5 27B model also increased the correlation coefficient from 0.25 to 0.49 but also increased sMAPE from 40.2% to 46.5%.

Table 5.2. Effect of synthetic fine-tuning on synthetic and real dataset evaluation. Fine-tuned values are shown with original model scores in parentheses.

		Gemma 4 31B		Qwen3.5 27B	
Parameter	**Metric**	**Syn**	**Real**	**Syn**	**Real**
DAP	r	0.90 (0.90)	0.90 (0.93)	0.47 (0.82)	0.54 (0.91)
	MAE	10.4 (11.4)	10.8 (8.7)	26.7 (14.4)	22.2 (9.3)
	sMAPE	18.9% (27.5%)	21.8% (23.3%)	45.4% (28.3%)	42.0% (20.8%)
Plant Count	r	0.43 (0.32)	0.17 (0.20)	0.25 (0.28)	0.13 (0.20)
	MAE	9.1 (10.0)	2.8 (2.7)	9.7 (16.0)	2.9 (3.0)
	sMAPE	43.3% (50.8%)	20.8% (20.0%)	48.6% (79.4%)	21.3% (22.8%)
Sun Elevation	r	0.64 (0.54)	0.33 (0.49)	0.38 (0.55)	0.04 (0.20)
	MAE	21.1 (18.0)	8.8 (7.4)	27.9 (22.6)	14.2 (10.6)
	sMAPE	90.4% (53.0%)	24.9% (16.0%)	121.3% (86.3%)	36.1% (21.7%)
Sun Azimuth	r	0.60 (0.56)	0.24 (0.53)	0.39 (0.57)	0.01 (-0.15)
	MAE	63.2 (64.5)	14.6 (8.6)	82.8 (66.4)	12.7 (14.5)
	sMAPE	59.9% (47.9%)	13.2% (7.9%)	51.4% (52.8%)	11.1% (13.8%)
Chlorophyll	r	0.77 (0.70)	0.73 (0.81)	0.61 (0.55)	0.23 (0.73)
	MAE	4.3 (5.2)	19.0 (11.8)	6.5 (5.9)	37.6 (10.9)
	sMAPE	32.1% (41.0%)	37.0% (29.7%)	50.3% (45.0%)	65.8% (27.0%)
Improvements / Total		12/15	1/15	4/15	5/15

Compared to synthetic few-shot examples, adding real-domain few-shot examples improved the average sMAPE from 45.4% to 19.4% for the Gemma 4 31B model and from 46.5% to 21.2% for the Qwen3.5 27B model. For Gemma 4 31B, the correlation coefficient also rose from 0.53 to 0.59 (Qwen3.5 27B: 0.49 to 0.38). The best overall result was from Gemma 4 31B with real few-shot, achieving a correlation coefficient of 0.59 and an sMAPE of 19.4%.

The DINOv3 VFM-based linear model could estimate the real dataset's DAP with a correlation coefficient of 0.72 and an MAE of 17.9 days with synthetic few-shot images. However, the VFM showed low correlation coefficients for plant count (0.04), sun azimuth (0.04), and chlorophyll (-0.39). When using real few-shot images in VFM linear model fittings, the average correlation coefficient increased from 0.15 to 0.46 and decreased the av-

Table 5.3. Effect of few-shot data modality for real dataset evaluation. DINOv3 embedding based minimum norm solution model was evaluated as a reference.

		Gemma 4 31B			Qwen3.5 27B			DINOv3 SAT	
Parameter	**Metric**	Zero (M2)	Syn (M4)	Real (M4)	Zero (M2)	Syn (M4)	Real (M4)	Syn (M4)	Real (M4)
DAP	r	0.81	0.88	0.93	0.88	0.86	0.91	0.72	0.75
	MAE	14.0	10.9	8.7	11.3	11.9	9.3	17.9	16.1
	sMAPE	35.2%	27.8%	23.3%	31.6%	30.4%	20.8%	43.9%	35.5%
Plant Count	r	0.20	0.10	0.20	0.09	0.14	0.20	0.04	0.01
	MAE	8.5	19.7	2.7	9.2	21.5	3.0	4.7	3.5
	sMAPE	42.0%	73.1%	20.0%	44.8%	53.2%	22.8%	30.0%	27.2%
Sun Elevation	r	0.40	0.54	0.49	-0.18	0.32	0.20	0.33	0.23
	MAE	10.9	13.0	7.4	13.3	17.4	10.6	20.6	7.6
	sMAPE	20.9%	25.4%	16.0%	25.0%	44.1%	21.7%	57.2%	16.5%
Sun Azimuth	r	0.28	0.53	0.53	0.27	0.46	-0.15	0.04	0.65
	MAE	38.6	109.7	8.6	70.0	108.2	14.5	49.7	14.6
	sMAPE	26.3%	58.9%	7.9%	49.1%	66.8%	13.8%	34.8%	15.0%
Chlorophyll	r	0.71	0.61	0.81	0.18	0.67	0.73	-0.39	0.67
	MAE	15.9	18.3	11.8	24.5	15.9	10.9	34.0	13.9
	sMAPE	37.8%	41.9%	29.7%	50.6%	37.9%	27.0%	112.2%	36.2%
Average	r	0.48	0.53	0.59	0.25	0.49	0.38	0.15	0.46
	sMAPE	32.4%	45.4%	19.4%	40.2%	46.5%	21.2%	55.6%	26.1%

erage sMAPE from 55.6% to 26.1% compared to synthetic few-shot fittings.

5.3.4 Visual Quality Evaluation

Fig. 5.6 and Fig. 5.7 show the generated cowpea plot images from the synthetic and real datasets. As shown in Fig. 5.7, the Gemma 4 31B model could reasonably estimate plant locations but had parameter estimation errors. For example, M1–M4 estimated larger plants than the input images, and brighter and yellowish plants because of chlorophyll content and solar position errors. Providing few-shot examples removed overestimated plants on 90 DAP from M1 to M3 but could not lower the average sMAPE substantially.

Similarly, tests on real images generally produced estimates of higher plant DAP, result-

ing in bigger plants on 10 and 28 DAP Fig. 5.7. Adding few-shot images (M4) improved the hallucinated plant locations on 28 DAP with M1 to M3, but the effect of few-shot images on later DAP was not visually recognizable. For the real dataset, adding more context from M1 to M5 decreased the sMAPE, showing an average sMAPE of 21.6% for this example.

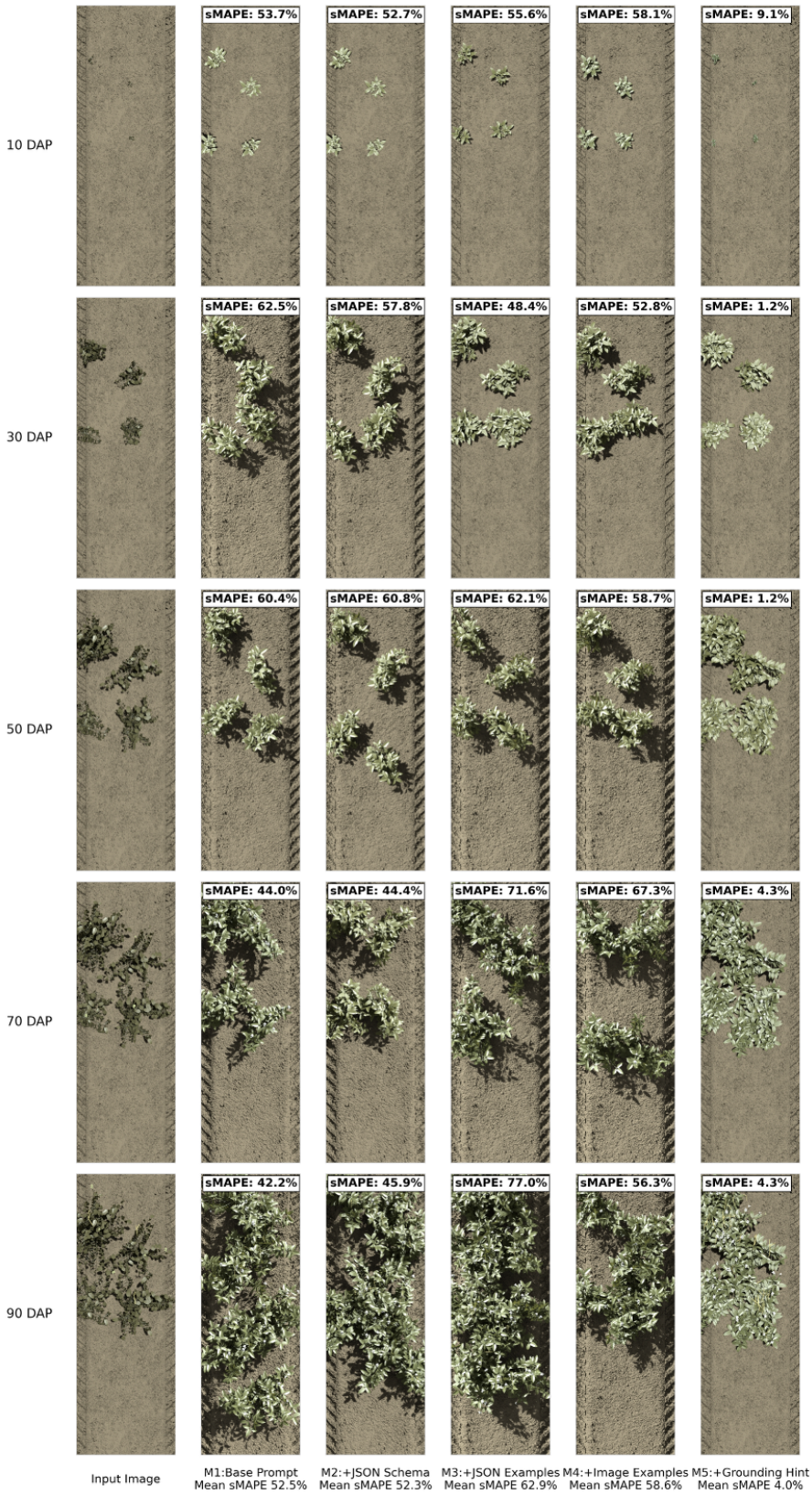


Figure 5.6. Examples of simulated cowpea plot generation results from the synthetic dataset based on in-context learning methods using the Gemma 4 31B model.

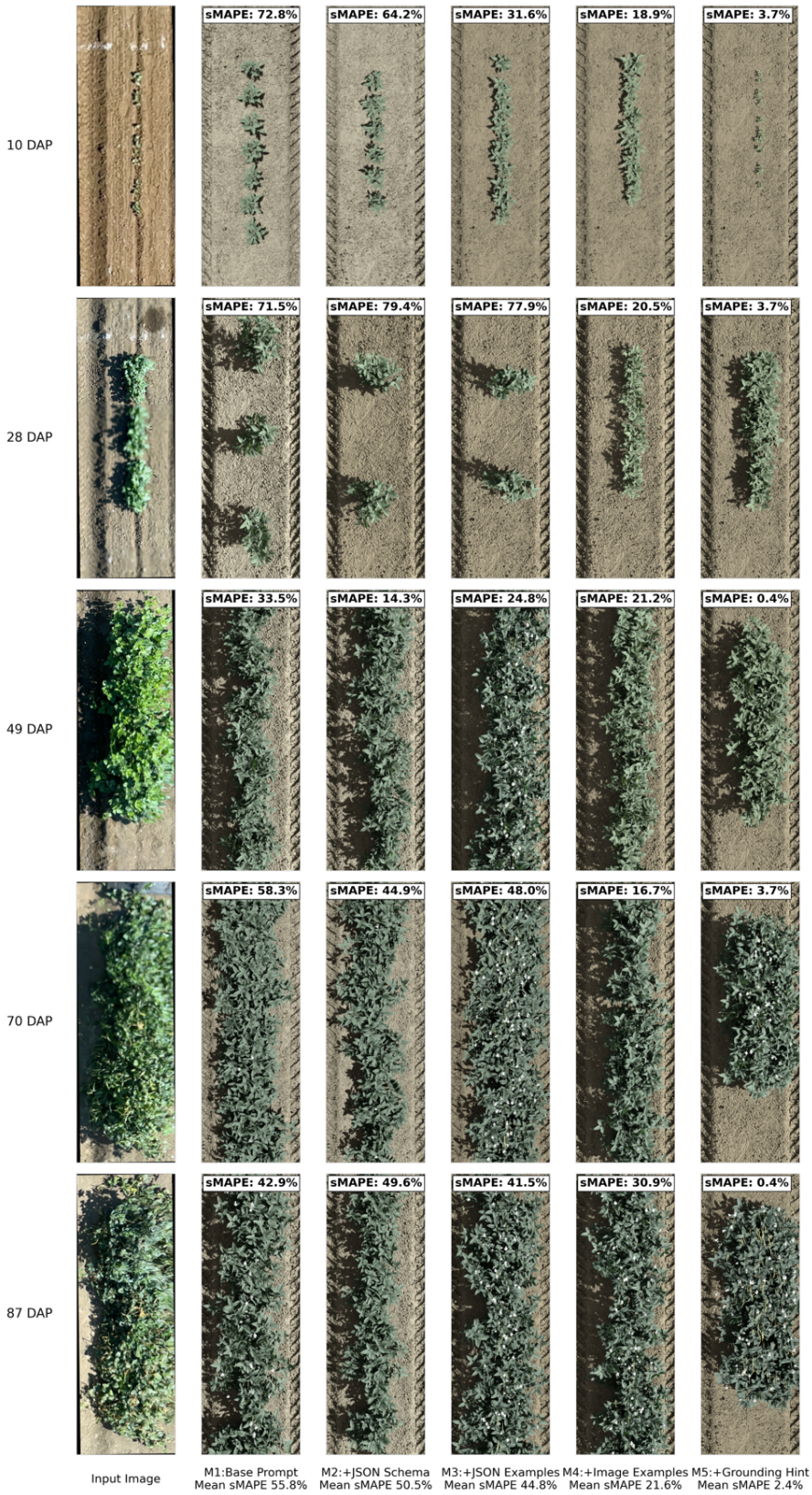


Figure 5.7. Examples of simulated cowpea plot generation results from the real dataset based on in-context learning methods using the Gemma 4 31B model.

5.4 Discussion

5.4.1 Structured Output Generation Performance

Forcing LLMs to respond in JSON format can be enabled by setting JSON mode in most LLM APIs, which provides a template and generates structured output that follows the given schema. Zhang et al. (2023) argued that structured decoding can improve tool-use performance by removing errors in JSON generation, even without fine-tuning the model. Beurer-Kellner et al. (2023); Geng et al. (2023) reported that structured output could not only improve the performance of classification tasks such as information extraction and named entity recognition, but also decrease the number of generated tokens. However, forcing an LLM to generate responses in a structured format reduced performance on complex reasoning tasks, such as mathematics, last-letter concatenation, and object shuffling (Tam et al., 2024). Iwanowski and Gahbler (2025) reported that forcing structured output in object localization tasks can lead to hallucinations, generating fake objects that conform to the desired structure.

In Fig. 5.3 (a) and (b), our results showed a maximum of 3.76% syntax error and 1.13% JSON key-missing error. Our benchmark shows that syntactic validity and semantic fidelity can diverge; for example, the Gemma 4 31B model showed 1.07% of JSON syntax errors when grounding hints were provided, even though this configuration showed the lowest errors and best visual quality. For practical deployment, structural correctness should therefore be treated as a prerequisite rather than as evidence of the semantic fidelity of the generated result, since perfect structured document generation performance is required for integrating the automated plant simulation generation pipeline into LLM agents and tool use (Yang et al., 2026; Zhang et al., 2023).

5.4.2 Effect of Model Size and Contextual Bias

The main finding of this study is that additional context does not reliably improve the in-context learning results. In several configurations, longer prompts increased error, because the models relied on contextual priors, copying values or distributions from the prompt, rather than extracting evidence from the target image. This effect was especially pronounced in smaller models, but it was not limited to them.

For example, Fig. 5.3 (c) shows that increasing model size occasionally increased DAP MAE for certain models, such as Qwen3.5 4B and 9B. This irregularity suggests that model size alone does not guarantee improved accuracy on zero-shot and few-shot agricultural tasks. Furthermore, additional context often increased MAEs, such as Fig. 5.3 (c), (d), (e), and (f), where DAP, plant count, plant location, and sun elevation errors increased after providing few-shot JSON or image examples. This suggests small models, such as Gemma 4 2B or Qwen3.5 9B, are unable to extract information from the target image reliably when the model's context increases, causing models to copy the values from the instruction prompt or few-shot examples, increasing the error metrics. The increase in error suggests that few-shot examples can act as noise when the model fails to capture a reliable visual signal and instead focuses on the wrong contextual patterns.

5.4.3 Effect of Fine-Tuning

Our fine-tuning results indicate that synthetic supervision alone is not sufficient to improve the robustness of simulation parameter estimation. As shown in Table 5.2, while Gemma 4 31B benefited modestly from fine-tuning on synthetic evaluation, these gains did not transfer to real-domain evaluations, and Qwen3.5 27B degraded after fine-tuning in both domains. This result suggests that fine-tuning can amplify the domain gap or disrupt capabilities that support zero-shot inference of VLMs. Luo et al. (2025) reported that continual fine-tuning can introduce catastrophic forgetting, particularly in larger models. Several studies reported the risk of performance degradation of multimodal large language

models during fine-tuning (Li et al., 2024a; Tang et al., 2025; Zhu et al., 2024). Further research is needed to find a better fine-tuning method to improve the accuracy of plot parameter estimation for both the synthetic and real domains without catastrophic forgetting.

5.4.4 Effect of Few-Shot Modality

As shown in Table 5.3, the modalities of the few-shot examples mattered substantially. Real few-shot examples generally outperformed synthetic few-shot examples on the real-dataset evaluation across both VLM families, indicating that prompt-domain alignment is important for maximizing in-context learning performance.

Similarly, the DINOv3 VFM showed better performance when the linear model was fitted from real few-shot examples and evaluated on the real dataset. However, the VFM fitted with synthetic few-shot examples could estimate DAP of real data but failed to estimate chlorophyll. Compared to VFMs' weak few-shot learning ability across domains, providing synthetic few-shot examples to VLMs led to performance improvement in DAP, solar positions, and chlorophyll.

This benefit will be useful when accurate ground truth is difficult to measure or inaccessible but synthetically guided context can be given, such as canopy photosynthesis, growth, water use, biomass estimation, and detailed plant architecture.

5.4.5 Integration with the Digital Twin Pipeline and Practical Applications

Digital twins simulate physical systems and update their states based on sensor data, supporting what-if experiments across crops and environments (Ganapathysubramanian et al., 2025; Verdouw et al., 2021). Our method bridges the gap between physiological metrics used by agronomists and morphological observations used by growers (Archambault et al., 2024), by simultaneously generating morphological traits (e.g., plant count, locations) and biophysical parameters, both encoded as JSON files and visualized through 3D plot simula-

tions.

This approach aligns with the multi-agent digital twin architecture proposed by Skobelev et al. (2020), where agents interact with agronomists through mobile interfaces to continuously recalculate yield forecasts and generate recommendations. Although our current results showed 19.4% average sMAPE for plant parameter estimation with few-shot examples Table 5.3, Gemma 4 31B with Real FS (M4), further improvements can be achieved by optimizing few-shot examples with accurately measured parameters and images. By combining in-context learning with precisely calibrated few-shot examples, we minimize sampling requirements to the few-shot scale (i.e., up to 5 samples) rather than the fine-tuning dataset scale (i.e., more than 1,000 samples).

5.5 Conclusion

We have introduced a benchmark for evaluating whether VLMs can generate structured plant simulation configurations from images and studied the effects of in-context learning methods, model family, model size, fine-tuning, and the modality of few-shot examples. The results show that current VLMs can produce syntactically valid JSON outputs and coarse parameter estimates, but performance remains inconsistent across model families and frequently below simple baselines for key geometric and biophysical parameters.

Domain-aligned few-shot examples improved results across both VLM families, yet models remained sensitive to contextual bias, often falling back on prompt priors rather than extracting visual evidence. Synthetic fine-tuning improved synthetic-domain performance but failed to transfer to real-domain evaluations, and in some cases degraded performance, suggesting that synthetic adaptation can amplify domain mismatch or disrupt zero-shot inference capabilities.

Therefore, we view image-to-simulation JSON generation as a promising yet challenging task for agricultural VLM research. Future work should explore stronger visual grounding

techniques, domain-invariant fine-tuning strategies, and structured decoding methods to improve JSON generation reliability. Expanding the benchmark to additional crop species, incorporating multispectral images, and evaluating under real-world deployment will further clarify the practical pathways for integrating VLMs into automated plant simulation pipelines.

Chapter 6

Conclusions

6.1 Overall Conclusions

From pixel-level thermal image upscaling to field-level plant architecture generation, this dissertation demonstrates how artificial intelligence (AI)-enabled sensing can overcome the scalability bottleneck of sensing for field-scale breeding trials by enhancing image resolution and generating plant simulation models from images.

The main conclusions drawn from the results presented in this dissertation are:

- RGB and thermal images can be aligned by converting the RGB image into a pseudo-thermal image and performing template matching against the thermal image. The aligned RGB and thermal images were fused using an SRGAN-based method that combines the RGB with low-resolution thermal inputs to create an upscaled thermal image output. The results showed that our algorithm could recover leaf details in low-resolution thermal images.
- A specialized plant architecture tokenization method was developed to convert an XML-based plant architecture file into a token sequence. A DINOv2- and GPT-2-based VLM was trained to generate a plant architecture token sequence from a synthetic input image. The results showed that plant architecture token sequences could be generated from input images, and that these could be rendered in a plant simulation program. This enables 3D organ-level understanding and generation of a plant architecture from RGB images.

- The plant architecture generation algorithm for a single plant was expanded to the plot scale using open-weight VLMs (2B to 31B parameters) and five in-context learning methods on synthetic and real datasets. The results showed that VLMs could generate plot-scale plant simulation configurations using optimized in-context learning prompts and few-shot examples without fine-tuning.

6.2 Future Work

Based on the results of this dissertation, the following directions for future work are recommended:

- To extend the multimodal image alignment approach to multiple modalities (e.g., thermal, RGB, and near-infrared (NIR)), each modality can be converted into a meta-domain shared across all modalities, such as shape masks or depth maps, and aligned in this shared domain. Further research should evaluate whether alignment in a shared meta-domain reduces registration errors compared to pairwise modality alignment.

- Image domain transfer and image super-resolution algorithms should be transitioned from GAN-based models to diffusion-based models, which have been shown to produce higher-quality results on domain transfer and upscaling tasks.

- The plant-level architecture generation algorithm developed in this study was trained on a single crop type (cowpea) using specific parameter distributions. Therefore, future studies are needed to extend the algorithm to different crop types with various parameter configurations.

- For the plot-level plant architecture generation algorithm, further testing with real datasets is needed to demonstrate its effectiveness in supporting plant breeders and scientists by providing detailed plant architecture models at the plot level.

- Depth images can improve 3D reconstruction accuracy, and multi-spectral data can improve the estimation of chemical compounds that determine leaf, flower, and fruit colors.

Chapter A

Appendix

A.1 Synthetic or Real Few-shot Data Effect on Real Data Evaluation with Ablation Study

Fig. A.1 shows Gemma 4 31B and Qwen3.5 27B models' evaluation result on the real image dataset when synthetic few-shot (Syn FS) or real few-shot (Real FS) examples were provided, and Real FS was provided, but the image was not provided for inference (Ablation).

Ablation study: To test whether the models relied on the target image rather than prompt priors alone, a blind ablation test was performed in which the final inference prompt omitted the target image and contained only the instruction 'Answer now:'. This experiment allows us to quantify how much of the generated output can be explained by contextual bias and models' hallucinations.

The blind ablation confirms that the apparent task performance can be partly decoupled from visual signals. When inference based on image context performs similarly to or worse than inference without an image, the output is being driven primarily by prompt statistics. For example, in Fig. A.1 some parameters showed higher error than ablation inference, such as Fig. A.1 (g) and (h), showing that the model could not reliably extract meaningful information from context.

In Fig. A.1 (g), the ablation inference showed lower error metrics than evaluations with real images by simply adhering to the distribution provided in the base prompt, showing

MAE values close to B_{med}. **JSON Integrity Evaluations:** As shown in Fig. A.1 (a), both Gemma 4 31B and Qwen3.5 27B models showed a syntax error rate below 1.97%, with no statistically significant difference across the few-shot example domains. In the blind ablation, Gemma 4 31B refused the task in M1–M4 by explicitly requesting the missing image, whereas both Gemma 4 31B in M5 and Qwen3.5 27B across multiple settings could still produce JSON outputs largely based on prompt priors, revealing distinct failure modes: refusal without visual evidence versus over-reliance on priors and hallucinations in answers. There were 1.96% key-missing errors in Syn FS for both models.

Geometric Evaluation: As shown in Fig. A.1 (d) M3, providing synthetic JSON examples to Gemma 4 31B model increased the DAP error but real JSON examples decreased the DAP error. In M4, providing either synthetic or real few-shot images decreased the MAE error, with real images producing a larger reduction than synthetic images. Qwen3.5 27B models showed similar DAP MAEs from M1 to M3, and lower MAE when real few-shot images were given at M4. Ablation inference showed MAEs close to B_{rnd}, showing that model was randomly guessing DAP when image was not provided.

Plant count evaluation on the real dataset with and without synthetic few-shot examples showed higher MAE than B_{rnd} for all models. In M3 and M4, providing real few-shot examples significantly decreased MAE, but still higher than B_{med}. Providing synthetic or real few-shot JSON increased plant count MAE for Qwen3.5 27B model but showed lower error when real few-shot images were given to the model. When real few-shot images were given in M4, it also decreased ablation MAE for Qwen3.5 27B. Gemma 4 31B model showed lower plant count MAE for M3 and M4 when real few-shot examples were given than synthetic few-shot.

Plant location Chamfer distance evaluation results on the real dataset showed that real few-shot JSON and images lowered the plant location error for M3 and M4 for both models. Qwen3.5 27B models showed higher or similar Chamfer distance than B_{rnd} for M1, M2

and M3, but lower than the baseline for M4 with real few-shot images. Ablation results on Qwen3.5 27B showed higher Chamfer distance for M1 and M2 and close to M3 for M4, suggesting the model hallucinated to generate median plant locations without seeing the target image.

Lighting and Biophysical Evaluations: As shown in Fig. A.1 (g) M3 – M5, real few-shot samples lowered the MAE error, but both models showed higher MAE than B_{med}. For Qwen3.5 27B model, ablation inference showed lowest MAE values across all the methods, showing the model could not estimate the sun elevation accurately. Real few-shot JSON and image decreased sun azimuth MAE in M3 and M4 for both models. However, the MAEs were higher than the B_{med} and close to ablation inference MAE. Chlorophyll MAE showed lower MAE when few-shot images were provided in M4, for both Gemma 4 31B and Qwen3.5 27B models. Grounding hint lowered MAE in M5 for Gemma 4 31B model, but Qwen3.5 27B remained within the range of confidence interval. Ablation inference for Chlorophyll showed MAEs that were higher than B_{med} and B_{rnd}.

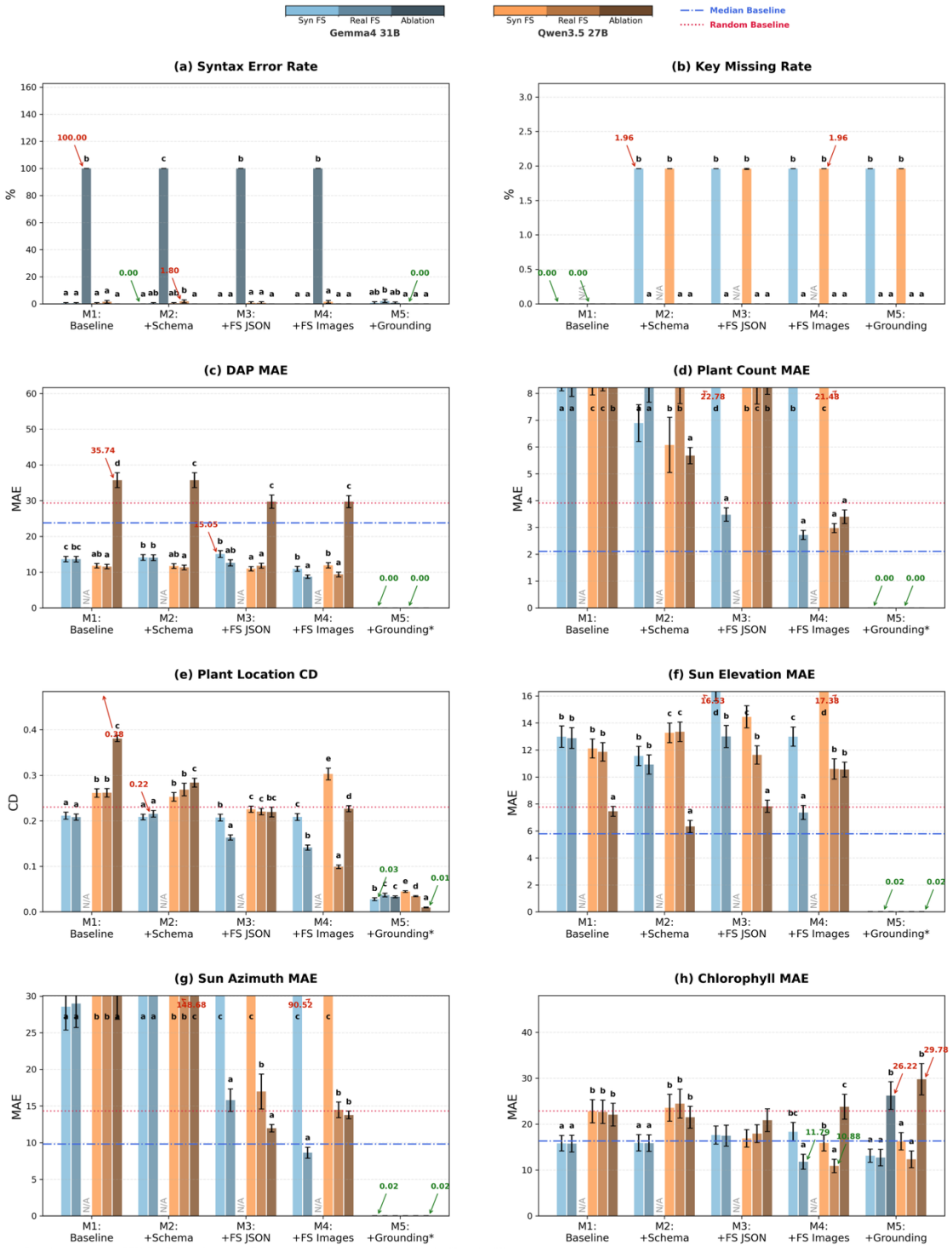


Figure A.1. Performance of Gemma 4 31B (blue) and Qwen3.5 27B (orange) under synthetic few-shot (Syn FS), real few-shot (Real FS), and blind-ablation conditions on the real dataset across five progressive in-context learning conditions (M1–M5). Panels (a–h) show syntax error rate, key-missing rate, DAP MAE, plant count MAE, plant location CD, sun elevation MAE, sun azimuth MAE, and chlorophyll MAE, respectively. Asterisks (*) denote exact values were included in M5 ground truth hint. Dash-dotted and dotted lines indicate median and random guess baselines, respectively. Ablation omits the target image at inference to test prompt priors against image evidence. Error bars represent 95% confidence intervals; letters indicate significant differences ($p < 0.05$).

Bibliography

Agrawal, A., Teney, D., Nematzadeh, A., 2022. Vision-Language Pretraining: Current Trends and the Future, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Association for Computational Linguistics, Dublin, Ireland. pp. 38–43. doi:10.18653/v1/2022.acl-tutorials.7.

Allen, M.T., Prusinkiewicz, P., DeJong, T.M., 2005. Using L-systems for modeling source–sink interactions, architecture and physiology of growing trees: The L-PEACH model. New Phytologist 166, 869–880. doi:10.1111/j.1469-8137.2005.01348.x.

Almasri, F., Debeir, O., 2018. RGB Guided Thermal Super-Resolution Enhancement, in: 2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech), pp. 1–5. doi:10.1109/CloudTech.2018.8713356.

Anderson, K.S., Hansen, C.W., Holmgren, W.F., Jensen, A.R., Mikofski, M.A., Driesse, A., 2023. Pvlib python: 2023 project update. Journal of Open Source Software 8, 5994. doi:10.21105/joss.05994.

Arar, M., Ginger, Y., Danon, D., Bermano, A.H., Cohen-Or, D., 2020. Unsupervised Multi-Modal Image Registration via Geometry Preserving Image-to-Image Translation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13410–13419.

Archambault, P., Sahraoui, H., Syriani, E., 2024. A Modeling Methodology for Crop Representation in Digital Twins for Smart Farming, in: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, ACM, Linz Austria. pp. 342–352. doi:10.1145/3652620.3688247.

Arshad, M.A., Jubery, T.Z., Roy, T., Nassiri, R., Singh, A.K., Singh, A., Hegde, C., Ganapathysubramanian, B., Balu, A., Krishnamurthy, A., Sarkar, S., 2025. Leveraging Vision Language Models for Specialized Agricultural Tasks, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, Tucson, AZ, USA. pp. 6320–6329. doi:10.1109/WACV61041.2025.00616.

Awais, M., Salem Abdulla Alharthi, A.H., Kumar, A., Cholakkal, H., Anwer, R.M., 2025. AgroGPT : Efficient Agricultural Vision-Language Model with Expert Tuning, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, Tucson, AZ, USA. pp. 5687–5696. doi:10.1109/WACV61041.2025.00555.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J. et al., 2023a. Qwen Technical Report. doi:10.48550/arXiv.2309.16609, arXiv:2309.16609.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J., 2023b. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. doi:10.48550/arXiv.2308.12966, arXiv:2308.12966.

Bailey, B.N., 2019. Helios: A Scalable 3D Plant and Environmental Biophysical Modeling Framework. Frontiers in Plant Science 10. doi:10.3389/fpls.2019.01185.

Bailey, B.N., 2025. A generalized framework for procedural generation of three-dimensional static and dynamic plant model geometries. doi:10.48550/arXiv.2512.17966, arXiv:2512.17966.

Bailey, B.N., Mahaffee, W.F., 2017. Rapid, high-resolution measurement of leaf area and leaf orientation using terrestrial LiDAR scanning data. Measurement Science and Technology 28, 064006. doi:10.1088/1361-6501/aa5cfd.

Baker, D.N., Bauer, F.M., Giraud, M., Schnepf, A., Göbbert, J.H., Scharr, H., Hvannberg, E.Þ., Riedel, M., 2024. A scalable pipeline to create synthetic datasets from functional–structural plant models for deep learning. in silico Plants 6, diad022. doi:10.1093/insilicoplants/diad022.

Bao, Y., Tang, L., Breitzman, M.W., Salas Fernandez, M.G., Schnable, P.S., 2019. Field-based robotic phenotyping of sorghum plant architecture using stereo vision. Journal of Field Robotics 36, 397–415. doi:10.1002/rob.21830.

Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C., 1977. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. Technical Report.

Bellvert, J., Pelechá, A., Pamies-Sans, M., Virgili, J., Torres, M., Casadesús, J., 2023. Assimilation of Sentinel-2 Biophysical Variables into a Digital Twin for the Automated Irrigation Scheduling of a Vineyard. Water 15, 2506. doi:10.3390/w15142506.

Beurer-Kellner, L., Fischer, M., Vechev, M., 2023. Prompting Is Programming: A Query Language for Large Language Models. Proceedings of the ACM on Programming Languages 7, 1946–1969. doi:10.1145/3591300.

Bhandari, M., 2016. Use of Infrared Thermal Imaging for Estimating Canopy Temperature in Wheat and Maize. Master's thesis. West Texas A&M University. arXiv:11310/133.

Blaya-Ros, P.J., Blanco, V., Domingo, R., Soto-Valles, F., Torres-Sánchez, R., 2020. Feasibility of Low-Cost Thermal Imaging for Monitoring Water Stress in Young and Mature Sweet Cherry Trees. Applied Sciences 10, 5461. doi:10.3390/app10165461.

Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., Godin, C., 2012. L-Py: An L-System Simulation Framework for Modeling Plant Architecture Development Based on a Dynamic Language. Frontiers in Plant Science 3. doi:10.3389/fpls.2012.00076.

Briechle, K., Hanebeck, U.D., 2001. Template matching using fast normalized cross correlation, in: Casasent, D.P., Chao, T.H. (Eds.), Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL. pp. 95–102. doi:10.1117/12.421129.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C. et al., 2020. Language Models are Few-Shot Learners. doi:10.48550/ARXIV.2005.14165.

Burgess, A.J., Retkute, R., Herman, T., Murchie, E.H., 2017. Exploring Relationships between Canopy Architecture, Light Distribution, and Photosynthesis in Contrasting Rice Genotypes Using 3D Canopy Reconstruction. Frontiers in Plant Science 8. doi:10.3389/fpls.2017.00734.

Cesco, S., Sambo, P., Borin, M., Basso, B., Orzes, G., Mazzetto, F., 2023. Smart agriculture and digital twins: Applications and challenges in a vision of sustainability. European Journal of Agronomy 146, 126809. doi:10.1016/j.eja.2023.126809.

Chebrolu, N., Magistri, F., Läbe, T., Stachniss, C., 2021. Registration of spatio-temporal point clouds of plants for phenotyping. PLOS ONE 16, e0247243. doi:10.1371/journal.pone.0247243.

Chen, X., Zhai, G., Wang, J., Hu, C., Chen, Y., 2016. Color guided thermal image super resolution, in: 2016 Visual Communications and Image Processing (VCIP), pp. 1–4. doi:10.1109/VCIP.2016.7805509.

Choi, H., Kim, Y., 2025. Multispectral-guided diffusion for enhancing thermal infrared image resolution, in: Matoba, O., Valenta, C.R., Shaw, J.A. (Eds.), SPIE Future Sensing Technologies 2025, SPIE, Yokohama, Japan. p. 16. doi:10.1117/12.3074612.

Cortés-Mendez, C., Hayet, J.B., 2024. Exploring the usage of diffusion models for thermal image super-resolution: A generic, uncertainty-aware approach for guided and non-guided schemes, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3123–3130. doi:10.1109/CVPRW63382.2024.00318.

D. M. Woebbecke, G. E. Meyer, K. Von Bargen, D. A. Mortensen, 1995. Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. Transactions of the ASAE 38, 259–269. doi:10.13031/2013.27838.

Dadrass Javan, F., Samadzadegan, F., Mehravar, S., Toosi, A., Khatami, R., Stein, A., 2021. A review of image fusion techniques for pan-sharpening of high-resolution satellite imagery. ISPRS Journal of Photogrammetry and Remote Sensing 171, 101–117. doi:10.1016/j.isprsjprs.2020.11.001.

Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L., 2019. Second-Order Attention Network for Single Image Super-Resolution, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA. pp. 11057–11066. doi:10.1109/CVPR.2019.01132.

DeepMind, G., 2026. Gemma 4 model card.

Dong, C., Loy, C.C., He, K., Tang, X., 2016. Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 295–307. doi:10.1109/TPAMI.2015.2439281.

Dong, Y., Ruan, C.F., Cai, Y., Lai, R., Xu, Z., Zhao, Y., Chen, T., 2024. XGrammar: Flexible and Efficient Structured Generation Engine for Large Language Models. doi:10.48550/ARXIV.2411.15100.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in: International Conference on Learning Representations (ICLR), arXiv. doi:10.48550/arXiv.2010.11929, arXiv:2010.11929.

Duchon, C.E., 1979. Lanczos Filtering in One and Two Dimensions. Journal of Applied Meteorology and Climatology 18, 1016–1022. doi:10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2.

Duvick, D.N., Smith, J.S.C., Cooper, M., 2003. Long-Term Selection in a Commercial Hybrid Maize Breeding Program, in: Plant Breeding Reviews. John Wiley & Sons, Ltd. chapter 4, pp. 109–151. doi:10.1002/9780470650288.ch4.

Dyrstad, J.S., Øye, E.R., 2025. Creating synthetic data sets for training of neural networks for automatic catch analysis in fisheries. Computers and Electronics in Agriculture 233, 110160. doi:10.1016/j.compag.2025.110160.

Escribà-Gelonch, M., Liang, S., van Schalkwyk, P., Fisk, I., Long, N.V.D., Hessel, V., 2024. Digital Twins in Agriculture: Orchestration and Applications. Journal of Agricultural and Food Chemistry 72, 10737–10752. doi:10.1021/acs.jafc.4c01934.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395. doi:10.1145/358669.358692.

Gaillard, M., Miao, C., Schnable, J.C., Benes, B., 2020. Voxel carving-based 3D reconstruction of sorghum identifies genetic determinants of light interception efficiency. Plant Direct 4, e00255. doi:10.1002/pld3.255.

Ganapathysubramanian, B., Sarkar, S., Singh, A., Singh, A.K., 2025. Digital twins for the plant sciences. Trends in Plant Science 30, 576–577. doi:10.1016/j.tplants.2024.12.013.

García-Tejero, I., Ortega-Arévalo, C., Iglesias-Contreras, M., Moreno, J., Souza, L., Tavira, S., Durán-Zuazo, V., 2018. Assessing the Crop-Water Status in Almond (Prunus dulcis Mill.) Trees via Thermal Imaging Camera Connected to Smartphone. Sensors 18, 1050. doi:10.3390/s18041050.

Geng, S., Josifoski, M., Peyrard, M., West, R., 2023. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning. doi:10.48550/ARXIV.2305.13971.

Giménez-Gallego, J., González-Teruel, J.D., Blaya-Ros, P.J., Toledo-Moreo, A.B., Domingo-Miguel, R., Torres-Sánchez, R., 2023. Automatic Crop Canopy Temperature Measurement Using a Low-Cost Image-Based Thermal Sensor: Application in a Pomegranate Orchard under a Permanent Shade Net House. Sensors 23, 2915. doi:10.3390/s23062915.

Giménez-Gallego, J., González-Teruel, J.D., Soto-Valles, F., Jiménez-Buendía, M., Navarro-Hellín, H., Torres-Sánchez, R., 2021. Intelligent thermal image-based sensor for affordable measurement of crop canopy temperature. Computers and Electronics in Agriculture 188, 106319. doi:10.1016/j.compag.2021.106319.

Golan, J.S., 2012. Moore–Penrose Pseudoinverses. Springer Netherlands, Dordrecht. pp. 441–452. doi:10.1007/978-94-007-2636-9_19.

Gonzalez-Dugo, V., Zarco-Tejada, P., Nicolás, E., Nortes, P.A., Alarcón, J.J., Intrigliolo, D.S., Fereres, E., 2013. Using high resolution UAV thermal imagery to assess the variability in the water status of five fruit tree species within a commercial orchard. Precision Agriculture 14, 660–678. doi:10.1007/s11119-013-9322-9.

Gupta, H., Mitra, K., 2020. Pyramidal Edge-maps and Attention based Guided Thermal Super-resolution. arXiv:2003.06216.

Gupta, H., Mitra, K., 2022. Toward Unaligned Guided Thermal Super-Resolution. IEEE Transactions on Image Processing 31, 433–445. doi:10.1109/TIP.2021.3130538.

Hägele, A., Bakouch, E., Kosson, A., Allal, L.B., Werra, L.V., Jaggi, M., 2024. Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations. doi:10.48550/arXiv.2405.18392, arXiv:2405.18392.

Han, D., Han, M., Unsloth team, 2023. Unsloth.

Hartley, Z.K.J., Stuart, L.A.G., French, A.P., Pound, M.P., 2025. PlantDreamer: Achieving Realistic 3D Plant Models with Diffusion-Guided Gaussian Splatting. doi:10.48550/arXiv.2505.15528, arXiv:2505.15528.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA. pp. 770–778. doi:10.1109/CVPR.2016.90.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2021. LoRA: Low-Rank Adaptation of Large Language Models. doi:10.48550/arXiv.2106.09685, arXiv:2106.09685.

Hufkens, K., Melaas, E.K., Mann, M.L., Foster, T., Ceballos, F., Robles, M., Kramer, B., 2019. Monitoring crop phenology using a smartphone based near-surface remote sensing approach. Agricultural and Forest Meteorology 265, 327–337. doi:10.1016/j.agrformet.2018.11.002.

Iseki, K., Olaleye, O., 2020. A new indicator of leaf stomatal conductance based on thermal imaging for field grown cowpea. Plant Production Science 23, 136–147. doi:10.1080/1343943X.2019.1625273.

Ishimwe, R., Abutaleb, K., Ahmed, F., 2014. Applications of Thermal Imaging in Agriculture—A Review. Advances in Remote Sensing 03, 128. doi:10.4236/ars.2014.33011.

Iwanowski, M., Gahbler, M., 2025. Multiple Large AI Models' Consensus for Object Detection—A Survey. Applied Sciences 15, 12961. doi:10.3390/app152412961.

Jacquemoud, S., Baret, F., 1990. PROSPECT: A model of leaf optical properties spectra. Remote Sensing of Environment 34, 75–91. doi:10.1016/0034-4257(90)90100-Z.

Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k., 2015. Spatial Transformer Networks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Kang, S., Moon, W., Kim, E., Heo, J.P., 2024. VLCounter: Text-Aware Visual Representation for Zero-Shot Object Counting. Proceedings of the AAAI Conference on Artificial Intelligence 38, 2714–2722. doi:10.1609/aaai.v38i3.28050.

Karwowski, R., Prusinkiewicz, P., 2004. The L-system-based plant-modeling environment L-studio 4.0, in: In Proceedings of the 4th International Workshop on Functional-Structural Plant Models, pp. pp. 403–405.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. doi:10.48550/arXiv.2308.04079, arXiv:2308.04079.

Khorsandi, A., Hemmat, A., Mireei, S.A., Amirfattahi, R., Ehsanzadeh, P., 2018. Plant temperature-based indices using infrared thermography for detecting water status in sesame under greenhouse conditions. Agricultural Water Management 204, 222–233. doi:10.1016/j.agwat.2018.04.012.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S., 2022. OCR-Free Document Understanding Transformer, in: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (Eds.), Computer Vision – ECCV 2022. Springer Nature Switzerland, Cham. volume 13688, pp. 498–517. doi:10.1007/978-3-031-19815-1_29.

Kim, J., Lee, J.K., Lee, K.M., 2016. Accurate Image Super-Resolution Using Very Deep Convolutional Networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA. pp. 1646–1654. doi:10.1109/CVPR.2016.182.

Kim, S., Heo, S., 2024. An agricultural digital twin for mandarins demonstrates the potential for individualized agriculture. Nature Communications 15, 1561. doi:10.1038/s41467-024-45725-x.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R., 2023. Segment Anything. doi:10.48550/arXiv.2304.02643, arXiv:2304.02643.

Klapp, I., Yafin, P., Oz, N., Brand, O., Bahat, I., Goldshtein, E., Cohen, Y., Alchanatis, V., Sochen, N., 2021. Computational end-to-end and super-resolution methods to improve thermal infrared remote sensing for agriculture. Precision Agriculture 22, 452–474. doi:10.1007/s11119-020-09746-y.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Kurth, W., Ong, Y., 2017. A design pattern in XL for implementing multiscale models, demonstrated with a fruit tree simulator. Acta Horticulturae , 35–42doi:10.17660/ActaHortic.2017.1160.6.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I., 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. doi:10.48550/arXiv.2309.06180, arXiv:2309.06180.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324. doi:10.1109/5.726791.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W., 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690.

Lee, K., Joshi, M., Turc, I., Hu, H., Liu, F., Eisenschlos, J., Khandelwal, U., Shaw, P., Chang, M.W., Toutanova, K., 2022. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. doi:10.48550/ARXIV.2210.03347.

Li, H., Ding, L., Fang, M., Tao, D., 2024a. Revisiting Catastrophic Forgetting in Large Language Model Tuning, in: Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA. pp. 4297–4308. doi:10.18653/v1/2024.findings-emnlp.249.

Li, J., Li, D., Xiong, C., Hoi, S., 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. doi:10.48550/arXiv.2201.12086, arXiv:2201.12086.

Li, J., Qi, X., Nabaei, S.H., Liu, M., Chen, D., Sun, Q., Zhang, X., Yin, X., Li, Z., 2025. A survey on 3D reconstruction techniques in plant phenotyping: From classical methods to Neural Radiance Fields (NeRF), 3D Gaussian Splatting (3DGS), and beyond. Plant Phenomics 7, 100137. doi:10.1016/j.plaphe.2025.100137.

Li, J., Qingqing, L., Qiao, J., Li, L., Yao, J., Tu, J., 2024b. Organ-Level Instance Segmentation of Oilseed Rape at Seedling Stage Based on 3D Point Cloud. Applied Engineering in Agriculture 40, 151–164. doi:10.13031/aea.15698.

Li, L., Zhang, Q., Huang, D., 2014. A Review of Imaging Techniques for Plant Phenotyping. Sensors (Basel, Switzerland) 14, 20078–20111. doi:10.3390/s141120078.

Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Honolulu, HI, USA. pp. 1132–1140. doi:10.1109/CVPRW.2017.151.

Lin, C.Y., 2004. ROUGE: A Package for Automatic Evaluation of Summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain. pp. 74–81.

Lindenmayer, A., 1968a. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. Journal of Theoretical Biology 18, 280–299. doi:10.1016/0022-5193(68)90079-9.

Lindenmayer, A., 1968b. Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. Journal of Theoretical Biology 18, 300–315. doi:10.1016/0022-5193(68)90080-5.

Liu, F., Eisenschlos, J., Piccinno, F., Krichene, S., Pang, C., Lee, K., Joshi, M., Chen, W., Collier, N., Altun, Y., 2023a. DePlot: One-shot visual language reasoning by plot-to-table translation, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada. pp. 10381–10399. doi:10.18653/v1/2023.findings-acl.660.

Liu, F., Piccinno, F., Krichene, S., Pang, C., Lee, K., Joshi, M., Altun, Y., Collier, N., Eisenschlos, J., 2023b. MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada. pp. 12756–12770. doi:10.18653/v1/2023.acl-long.714.

Liu, H., Li, C., Wu, Q., Lee, Y.J., 2023c. Visual Instruction Tuning. doi:10.48550/arXiv.2304.08485, arXiv:2304.08485.

Lobet, G., Pound, M.P., Diener, J., Pradal, C., Draye, X., Godin, C., Javaux, M., Leitner, D., Meunier, F., Nacry, P., Pridmore, T.P., Schnepf, A., 2015. Root System Markup Language: Toward a Unified Root Architecture Description Language. Plant Physiology 167, 617–627. doi:10.1104/pp.114.253625.

Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., Zhang, Y., 2025. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning. IEEE Transactions on Audio, Speech and Language Processing 33, 3776–3786. doi:10.1109/TASLPRO.2025.3606231.

Mandanici, E., Tavasci, L., Corsini, F., Gandolfi, S., 2019. A multi-image super-resolution algorithm applied to thermal imagery. Applied Geomatics 11, 215–228. doi:10.1007/s12518-019-00253-y.

Mangus, D.L., Sharda, A., Zhang, N., 2016. Development and evaluation of thermal infrared imaging system for high spatial and temporal resolution crop water stress monitoring of corn within a greenhouse. Computers and Electronics in Agriculture 121, 149–159. doi:10.1016/j.compag.2015.12.007.

Marius, M., David G, L., 2009. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, in: Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, SciTePress - Science and and Technology Publications, Lisboa, Portugal. pp. 331–340. doi:10.5220/0001787803310340.

Messina, G., Modica, G., 2020. Applications of UAV Thermal Imagery in Precision Agriculture: State of the Art and Future Research Outlook. Remote Sensing 12, 1491. doi:10.3390/rs12091491.

Meyer, L., Gilson, A., Schmidt, U., Stamminger, M., 2024. FruitNeRF: A Unified Neural Radiance Field based Fruit Counting Framework. arXiv:2408.06190.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2022. NeRF: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65, 99–106. doi:10.1145/3503250.

Minkina, W., Dudzik, S., 2009. Infrared Thermography: Errors and Uncertainties. 1 ed., Wiley. doi:10.1002/9780470682234.

Möller, M., Alchanatis, V., Cohen, Y., Meron, M., Tsipris, J., Naor, A., Ostrovsky, V., Sprintsin, M., Cohen, S., 2007. Use of thermal and visible imagery for estimating crop water status of irrigated grapevine*. Journal of Experimental Botany 58, 827–838. doi:10.1093/jxb/erl115.

Moser, B.B., Shanbhag, A.S., Raue, F., Frolov, S., Palacio, S., Dengel, A., 2024. Diffusion Models, Image Super-Resolution And Everything: A Survey. https://arxiv.org/abs/2401.00736v3. doi:10.1109/TNNLS.2024.3476671.

Nelder, J.A., Mead, R., 1965. A Simplex Method for Function Minimization. The Computer Journal 7, 308–313. doi:10.1093/comjnl/7.4.308.

Niinemets, Ü., 2010. A review of light interception in plant stands from leaf to canopy in different plant functional types and in species with varying shade tolerance. Ecological Research 25, 693–714. doi:10.1007/s11284-010-0712-4.

Ojo, T., La, T., Morton, A., Stavness, I., 2024. Splanting: 3D plant capture with gaussian splatting, in: SIGGRAPH Asia 2024 Technical Communications, ACM, Tokyo Japan. pp. 1–4. doi:10.1145/3681758.3698009.

OpenAI, Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A.J., Welihinda, A., Hayes, A., Radford, A., Mądry, A., Baker-Whitcomb, A., Beutel, A., Borzunov, A., Carney, A., Chow, A., Kirillov, A., Nichol, A., Paino, A. et al., 2024. GPT-4o System Card. doi:10.48550/arXiv.2410.21276, arXiv:2410.21276.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G. et al., 2024. DINOv2: Learning Robust Visual Features without Supervision. doi:10.48550/arXiv.2304.07193, arXiv:2304.07193.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2001. BLEU: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, Association for Computational Linguistics, Philadelphia, Pennsylvania. p. 311. doi:10.3115/1073083.1073135.

Parihar, G., Saha, S., Giri, L.I., 2021. Application of infrared thermography for irrigation scheduling of horticulture plants. Smart Agricultural Technology 1, 100021. doi:10.1016/j.atech.2021.100021.

Parry, C., Blonquist Jr., J.M., Bugbee, B., 2014. In situ measurement of leaf chlorophyll concentration: Analysis of the optical/absolute relationship. Plant, Cell & Environment 37, 2508–2520. doi:10.1111/pce.12324.

Peladarinos, N., Piromalis, D., Cheimaras, V., Tserepas, E., Munteanu, R.A., Papageorgas, P., 2023. Enhancing Smart Agriculture by Implementing Digital Twins: A Comprehensive Review. Sensors (Basel, Switzerland) 23, 7128. doi:10.3390/s23167128.

Pele, F.D., Yeboah, A., Buari, M.S., Kofi, E.J., 2016. Morpho-physiological parameters used in selecting drought tolerant cowpea varieties using drought index .

QwenTeam, 2026. Qwen3.5: Towards Native Multimodal Agents.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Raimundo, J., Lopez-Cuervo Medina, S., Prieto, J.F., Aguirre de Mata, J., 2021. Super Resolution Infrared Thermal Imaging Using Pansharpening Algorithms: Quantitative Assessment and Application to UAV Thermal Imaging. Sensors 21, 1265. doi:10.3390/s21041265.

Rivadeneira, R., Sappa, A., Vintimilla, B., 2022. Multi-Image Super-Resolution for Thermal Images, in: Proceedings of the 17th International Joint Conference on Computer

Vision, Imaging and Computer Graphics Theory and Applications, SciTePress. pp. 635–642. doi:10.5220/0010899500003124.

Rivadeneira, R.E., Sappa, A.D., Vintimilla, B.X., Nathan, S., Kansal, P., Mehri, A., Behjati Ardakani, P., Dalal, A., Akula, A., Sharma, D., Pandey, S., Kumar, B., Yao, J., Wu, R., Feng, K., Li, N., Zhao, Y., Patel, H., Chudasama, V., Prajapati, K. et al., 2021. Thermal Image Super-Resolution Challenge - PBVS 2021, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4354–4362. doi:10.1109/CVPRW53098.2021.00492.

Saeed, F., Sun, J., Ozias-Akins, P., Chu, Y.J., Li, C.C., 2023. PeanutNeRF: 3D Radiance Field for Peanuts, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Vancouver, BC, Canada. pp. 6254–6263. doi:10.1109/CVPRW59228.2023.00665.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M., 2021. Image Super-Resolution via Iterative Refinement. https://arxiv.org/abs/2104.07636v2.

Salter, W.T., Shrestha, A., Barbour, M.M., 2021. Open source 3D phenotyping of chickpea plant architecture across plant development. Plant Methods 17, 95. doi:10.1186/s13007-021-00795-6.

Sarvaiya, J., Patnaik, S., Bombaywala, S., 2009. Image Registration by Template Matching Using Normalized Cross-Correlation, in: 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 819–822. doi:10.1109/ACT.2009.207.

Schiller, C., Schmidtlein, S., Boonman, C., Moreno-Martínez, A., Kattenborn, T., 2021. Deep learning and citizen science enable automated plant trait predictions from photographs. Scientific Reports 11, 16395. doi:10.1038/s41598-021-95616-0.

Schnepf, A., Leitner, D., Landl, M., Lobet, G., Mai, T.H., Morandage, S., Sheng, C., Zörner, M., Vanderborght, J., Vereecken, H., 2018. CRootBox: A structural–functional modelling framework for root systems. Annals of Botany 121, 1033–1053. doi:10.1093/aob/mcx221.

Shen, P., Jing, X., Deng, W., Jia, H., Wu, T., 2025. PlantGaussian: Exploring 3D Gaussian splatting for cross-time, cross-scene, and realistic 3D plant visualization and beyond. The Crop Journal 13, 607–618. doi:10.1016/j.cj.2025.01.011.

Shinoda, R., Inoue, N., Kataoka, H., Onishi, M., Ushiku, Y., 2025. AgroBench: Vision-Language Model Benchmark in Agriculture. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) , 7634–7644.

Sievänen, R., Godin, C., DeJong, T.M., Nikinmaa, E., 2014. Functional–structural plant models: A growing paradigm for plant studies. Annals of Botany 114, 599–603. doi:10.1093/aob/mcu175.

Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J. et al., 2025. DINOv3. doi:10.48550/arXiv.2508.10104, arXiv:2508.10104.

Singh, B., Kumar, S., Elangovan, A., Vasht, D., Arya, S., Duc, N.T., Swami, P., Pawar, G.S., Raju, D., Krishna, H., Sathee, L., Dalal, M., Sahoo, R.N., Chinnusamy, V., 2023. Phenomics based prediction of plant biomass and leaf area in wheat using machine learning approaches. Frontiers in Plant Science 14, 1214801. doi:10.3389/fpls.2023.1214801.

Skobelev, P., Laryukhin, V., Simonova, E., Goryanin, O., Yalovenko, V., Yalovenko, O., 2020. Multi-agent approach for developing a digital twin of wheat, in: 2020 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, Bologna, Italy. pp. 268–273. doi:10.1109/SMARTCOMP50058.2020.00062.

Smith, D.T., Potgieter, A.B., Chapman, S.C., 2021. Scaling up high-throughput phenotyping for abiotic stress selection in the field. Theoretical and Applied Genetics 134, 1845–1866. doi:10.1007/s00122-021-03864-5.

Smith, P.R., 1981. Bilinear interpolation of digital images. Ultramicroscopy 6, 201–204. doi:10.1016/0304-3991(81)90061-9.

Soualiou, S., Wang, Z., Sun, W., de Reffye, P., Collins, B., Louarn, G., Song, Y., 2021. Functional–Structural Plant Models Mission in Advancing Crop Science: Opportunities and Prospects. Frontiers in Plant Science 12. doi:10.3389/fpls.2021.747142.

Subramanian, S., Merrill, W., Darrell, T., Gardner, M., Singh, S., Rohrbach, A., 2022. ReCLIP: A Strong Zero-Shot Baseline for Referring Expression Comprehension, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland. pp. 5198–5215. doi:10.18653/v1/2022.acl-long.357.

Tam, Z.R., Wu, C.K., Tsai, Y.L., Lin, C.Y., Lee, H.y., Chen, Y.N., 2024. Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models. doi:10.48550/arXiv.2408.02442, arXiv:2408.02442.

Tang, L., Tian, Z., Li, K., He, C., Zhou, H., Zhao, H., Li, X., Jia, J., 2025. Mind the Interference: Retaining Pre-trained Knowledge in Parameter Efficient Continual Learning of Vision-Language Models, in: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (Eds.), Computer Vision – ECCV 2024. Springer Nature Switzerland, Cham. volume 15094, pp. 346–365. doi:10.1007/978-3-031-72764-1_20.

Tattersall, G.J., 2016. Infrared thermography: A non-invasive window into thermal physiology. Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology 202, 78–98. doi:10.1016/j.cbpa.2016.02.022.

Teichmann, T., Muhr, M., 2015. Shaping plant architecture. Frontiers in Plant Science 6. doi:10.3389/fpls.2015.00233.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G., 2023. LLaMA: Open and Efficient Foundation Language Models. doi:10.48550/arXiv.2302.13971, arXiv:2302.13971.

Truong, S.K., McCormick, R.F., Rooney, W.L., Mullet, J.E., 2015. Harnessing Genetic Variation in Leaf Angle to Increase Productivity of Sorghum bicolor. Genetics 201, 1229–1238. doi:10.1534/genetics.115.178608.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ukasz Kaiser, Ł., Polosukhin, I., 2017. Attention is All you Need, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Verbeeck, H., Bauters, M., Jackson, T., Shenkin, A., Disney, M., Calders, K., 2019. Time for a Plant Structural Economics Spectrum. Frontiers in Forests and Global Change 2, 43. doi:10.3389/ffgc.2019.00043.

Verdouw, C., Tekinerdogan, B., Beulens, A., Wolfert, S., 2021. Digital twins in smart farming. Agricultural Systems 189, 103046. doi:10.1016/j.agsy.2020.103046.

Wang, L., Jin, T., Yang, J., Leonardis, A., Wang, F., Zheng, F., 2024a. Agri-LLaVA: Knowledge-Infused Large Multimodal Assistant on Agricultural Pests and Diseases. doi:10.48550/arXiv.2412.02158, arXiv:2412.02158.

Wang, X., Xie, L., Dong, C., Shan, Y., 2021. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data, in: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), IEEE, Montreal, BC, Canada. pp. 1905–1914. doi:10.1109/ICCVW54120.2021.00217.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X., 2018. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. doi:10.48550/arXiv.1809.00219, arXiv:1809.00219.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13, 600–612. doi:10.1109/TIP.2003.819861.

Wang, Z., Lorraine, J., Wang, Y., Su, H., Zhu, J., Fidler, S., Zeng, X., 2024b. LLaMA-Mesh: Unifying 3D Mesh Generation with Language Models. doi:10.48550/arXiv.2411.09595, arXiv:2411.09595.

Williams, R.J., Zipser, D., 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation 1, 270–280. doi:10.1162/neco.1989.1.2.270.

Yan, C., Xiang, J., Qin, L., Wang, B., Shi, Z., Xiao, W., Hayat, M., Qiu, G.Y., 2023. High temporal and spatial resolution characteristics of evaporation, transpiration, and evapotranspiration from a subalpine wetland by an advanced UAV technology. Journal of Hydrology 623, 129748. doi:10.1016/j.jhydrol.2023.129748.

Yang, B., Chen, Y., Feng, L., Zhang, Y., Xu, X., Zhang, J., Aierken, N., Huang, R., Lin, H., Ying, Y., Li, S., 2025. AgriGPT-VL: Agricultural Vision-Language Understanding Suite. doi:10.48550/arXiv.2510.04002, arXiv:2510.04002.

Yang, J., Jiang, D., He, L., Siu, S., Zhang, Y., Liao, D., Li, Z., Zeng, H., Jia, Y., Wang, H., Schneider, B., Ruan, C., Ma, W., Lyu, Z., Wang, Y., Lu, Y., Do, Q.D., Jiang, Z., Nie, P., Chen, W., 2026. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. doi:10.48550/arXiv.2505.20139, arXiv:2505.20139.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H., 2024. Depth Anything V2. doi:10.48550/arXiv.2406.09414, arXiv:2406.09414.

Yun, H., Lo, S., Diepenbrock, C.H., Bailey, B.N., Earles, J.M., 2024. VisTA-SR: Improving the Accuracy and Resolution of Low-Cost Thermal Imaging Cameras for Agriculture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470–5479.

Yun, H., Uyehara, I.K., Droutsas, I., Ranario, E., Diepenbrock, C.H., Bailey, B.N., Earles, J.M., 2026a. A vision language model for generating XML-based organ-level plant architecture representations of cowpea from simulated images. Frontiers in Artificial Intelligence 9, 1844338. doi:10.3389/frai.2026.1844338.

Yun, H., Uyehara, I.K., Ranario, E., Lundqvist, L., Diepenbrock, C.H., Bailey, B.N., Earles, J.M., 2026b. Using Vision Language Foundation Models to Generate Plant Simulation Configurations via In-Context Learning. doi:10.48550/arXiv.2603.08930, arXiv:2603.08930.

Yun, K., Kim, S.H., 2023. Cropbox: A declarative crop modelling framework. in silico Plants 5, diac021. doi:10.1093/insilicoplants/diac021.

Zhang, J., Zhang, Z., Neng, F., Xiong, S., Wei, Y., Cao, R., Wei, Q., Ma, X., Wang, X., 2022. Canopy light distribution effects on light use efficiency in wheat and its mechanism. Frontiers in Ecology and Evolution 10. doi:10.3389/fevo.2022.1023117.

Zhang, K., Chen, H., Li, L., Wang, W., 2023. Don't Fine-Tune, Decode: Syntax Error-Free Tool Use via Constrained Decoding. doi:10.48550/ARXIV.2310.07075.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y., 2018. Residual Dense Network for Image Super-Resolution, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT. pp. 2472–2481. doi:10.1109/CVPR.2018.00262.

Zheng, C., Abd-Elrahman, A., Whitaker, V.M., Dalid, C., 2022. Deep Learning for Strawberry Canopy Delineation and Biomass Prediction from High-Resolution Images. Plant Phenomics 2022. doi:10.34133/2022/9850486.

Zhou, X.R., Schnepf, A., Vanderborght, J., Leitner, D., Lacointe, A., Vereecken, H., Lobet, G., 2020. CPlantBox, a whole-plant modelling framework for the simulation of water- and carbon-related processes. in silico Plants 2, diaa001. doi:10.1093/insilicoplants/diaa001.

Zhou, Y., Yan, H., Ding, K., Cai, T., Zhang, Y., 2024. Few-Shot Image Classification of Crop Diseases Based on Vision–Language Models. Sensors 24, 6109. doi:10.3390/s24186109.

Zhu, B., Zhang, Y., Sun, Y., Shi, Y., Ma, Y., Guo, Y., 2023. Quantitative estimation of organ-scale phenotypic parameters of field crops through 3D modeling using extremely low altitude UAV images. Computers and Electronics in Agriculture 210, 107910. doi:10.1016/j.compag.2023.107910.

Zhu, D., Sun, Z., Li, Z., Shen, T., Yan, K., Ding, S., Kuang, K., Wu, C., 2024. Model Tailor: Mitigating Catastrophic Forgetting in Multi-modal Large Language Models. doi:10.48550/ARXIV.2402.12048.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232.

Zia, S., Du, W., Spreer, W., Spohrer, K., He, X., Muller, J., 2012. Assessing crop water stress of winter wheat by thermography under different irrigation regimes in North China Plain. International Journal of Agricultural and Biological Engineering 5, 24–34. doi:10.25165/ijabe.v5i3.573.

Ziamtsov, I., Navlakha, S., 2019. Machine Learning Approaches to Improve Three Basic Plant Phenotyping Tasks Using Three-Dimensional Point Clouds. Plant Physiology 181, 1425–1440. doi:10.1104/pp.19.00524.

Ziamtsov, I., Navlakha, S., 2020. Plant 3D (P3D): A plant phenotyping toolkit for 3D point clouds. Bioinformatics 36, 3949–3950. doi:10.1093/bioinformatics/btaa220.

Zou, J., Gao, B., Song, Y., Qin, J., 2022. A review of deep learning-based deformable medical image registration. Frontiers in Oncology 12. doi:10.3389/fonc.2022.1047215.