LAESI: Leaf Area Estimation with Synthetic Imagery

Jacek Kałużny
AMU
jacek.kaluzny@amu.edu.pl

Yannik Schreckenberg
TUM
yannik.schreckenberg@tum.de

Karol Cyganik
AMU
karol.cyganik@amu.edu.pl

Peter Annighöfer
TUM
peter.annighoefer@tum.de

Sören Pirk
GreenMatterAI / CAU
soeren.pirk@greenmatter.ai

Dominik L. Michels
GreenMatterAI / KAUST / TU Darmstadt
dominik.michels@greenmatter.ai

Mikolaj Cieslak
GreenMatterAI
mikolaj.cieslak@greenmatter.ai

Farhah Assaad-Gerbert
TUM
farhah.assaad@tum.de

Bedrich Benes
Purdue University
bbenes@purdue.edu

Wojciech Pałubicki
GreenMatterAI / AMU
wojciech.palubicki@greenmatter.ai

Abstract

We introduce LAESI, a Synthetic Leaf Dataset of 100K synthetic leaf images on millimeter paper, each with semantic masks and surface area labels. This dataset provides a resource for leaf morphology analysis aimed at beech and oak leaves. We evaluate the applicability of the dataset by training machine learning models for leaf surface area prediction and semantic segmentation, using real images for validation. Our validation shows that these models can be trained to predict leaf surface area with a relative error not greater than an average human annotator. LAESI also provides an efficient framework based on 3D procedural models and generative AI for the large-scale, controllable generation of data with potential further applications in agriculture and biology. We evaluate the inclusion of generative AI in our procedural data generation pipeline and show how data filtering based on annotation consistency results in datasets allows training the highest performing vision models.

1. Introduction

In recent years, the integration of machine vision algorithms in agriculture has been instrumental in enhancing productivity and sustainability. One limitation of the machine learning algorithms is their reliance on accurate and extensive training data. This is because acquiring sufficiently annotated real-world data, particularly for tasks like leaf analysis, is often costly and time-consuming [22].

We present LAESI, a Synthetic Leaf Dataset, and two procedural models for its generation. The first is a procedural model for millimeter paper, and the other is for leaf shape generation. Our method employs ControlNet [26] to improve the visual realism of renderings similarly to Anagnostopoulou et al. [1]. Using computationally efficient procedural models paired with generative AI models allows for a fully automatic, controllable, and large-scale generation of synthetic data that is useful for training deep learning models for vision tasks.

The LAESI dataset provides annotations consisting of semantic masks and surface area labels. We demonstrate the utility of this dataset by training machine vision models on leaf surface area prediction and semantic segmentation. Furthermore, we compare vision models trained with different blends of synthetic data and real data against a baseline model trained on 1,7K real annotated images.

Our pipeline for LAESI generates realistic and diverse synthetic leaf images through several stages (Fig. 1): *(1) Procedural Generation of Millimeter Paper Background:* generate paper textures and grid alignments to for consis-

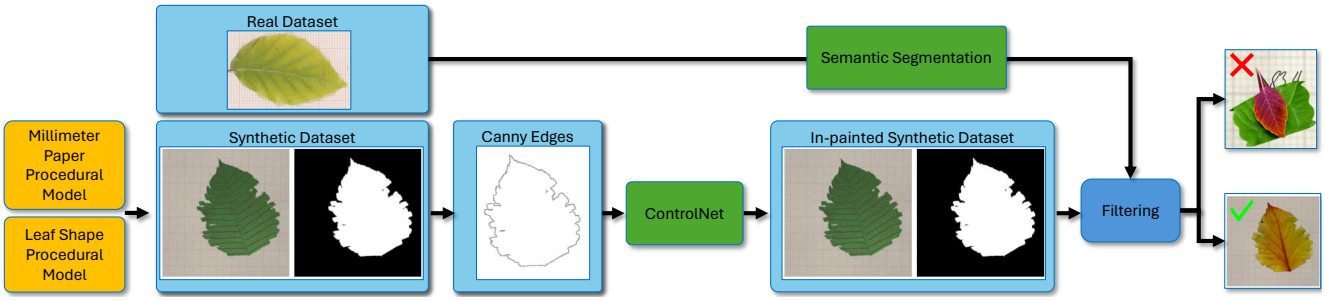

Figure 1. A model of the LAESI pipeline: *Procedural Generation of Millimeter Paper Background and Leaf Shape* generates diverse paper textures, grid alignments, and a range of leaf shapes, sizes, and textures. *Rendering and Final Synthetic Dataset Composition* combines leaves with the background with realistic lighting and generates annotations such as semantic masks, surface area labels, and canny edges. *Dataset Inpainting* utilizing the ControlNet-based pipeline for inpainting of Canny edges generates leaf images inside the masked regions of data points. *Dataset Filtering* discards the leaf data points with inpainting results that reduce consistency with their annotations by using a semantic segmentation model.

tent scale reference; *(2) Leaf Shape and Texturing Procedural Model:* generates a range of leaf shapes, sizes, and textures to increase the datasets variability; *(3) Semantic Mask and Surface Area Labeling:* Semantic masks delineate leaf boundaries against the millimeter paper, together with accurate surface area labels; *(4) Rendering and Final Image Composition:* the synthetic leaves are combined with the millimeter paper background, adding realistic lighting, shadow effects, and overall image composition; *(5) Dataset Inpainting:* Each image in the dataset is processed using a ControlNet-based [26] pipeline for accurate inpainting of leaf masks using canny edges and text prompts as input; *(6) Semantic Segmentation-based Quality Control:* Inpainted synthetic images are semantically segmented into leaf and background and undergo a comparison with the procedurally generated ground truth masks to establish annotation consistency.

2. Related Work

Deep learning neural models have shown strong progress in many areas, but they require a large volume of high-quality data for effective training. While data are abundant, annotated data are expensive and difficult to obtain, especially in natural sciences, where the variance of a single biological species can be significantly high in both shape and appearance (texture). Various approaches have been developed to address this challenge, including semi-supervised, self-supervised learning, and synthetic data generation.

One notable approach in synthetic data generation is DatasetGAN [27], which proposed a pipeline involving initial image generation by StyleGAN, followed by manual annotation of a few images for a specific task, and then training a small model to produce similar segmentation masks from StyleGAN features. This method allows the generation of a large number of labeled images with minimal manual effort. BigDatasetGAN [27] extended this concept using BigGAN for generating a diverse range of images, scaling it to the complexity of datasets like ImageNet.

In the realm of complex scene generation, Yang et al. [24] proposed a method for image generation based on specific layouts, compressing RGB images into patch tokens and utilizing a Transformer with Focal Attention. Sun et al. [19] introduced SHIFT, a synthetic driving dataset with variations in weather, time of day, and densities of vehicles and pedestrians, using domain adaptation for realistic simulations. Yan et al. [23] developed a visual localization system using a synthetic data generation tool that blends real and synthetic worlds, generating data with multiple annotations.

Similarly, Anagnostopoulou et al. [1] developed a Realistic Synthetic Mushroom Scenes Dataset for use in mushroom harvesting robotics. Close to our approach is the work of Ubbens et al. [20], who developed a synthetic leaf model of rosette plants for counting focusing on describing the whole plant morphology. Our method presents a targeted approach for the generation of a large dataset of leaves for various tree species. Furthermore, Zhang et al. [26] demonstrated advancements in text-to-image diffusion models by adding conditional control.

Leaf appearance and modeling have been studied by computer graphics for decades. Chiba et al. [3] proposed a method for leaf coloring and arrangement. Wang et al. used physics to simulate leaf growth in [21] and leaf venation patterns have been studied in [8, 16]. A general approach for leaf shape development considering experimental data from developmental biology was proposed in [17]. We are not aware of any systematic approach to the generation of large leaf datasets for deep learning.

Our work builds on existing research by investigating the integration of efficient, controllable 3D procedural models similar to ones used in Raistrick et al. [14] into a pipeline leveraging generative AI models to train deep learning mod-

els for specialized vision tasks where annotated real-world data are scarce or expensive to acquire.

3. Method

LAESI allows for fully automatic large-scale synthetic data generation by leveraging simple but computationally efficient 3D procedural models for rendering. We implemented these procedural models with Unity. In this section, we discuss the individual components of the LAESI modeling and rendering pipeline.

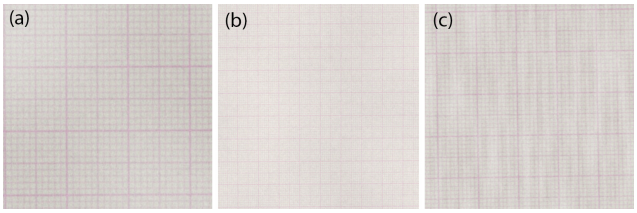

Figure 2. Selection of different millimeter papers generated using our procedural shader method ranging from sharp to blurry.

3.1. Procedural Millimeter Paper Model

The procedural millimeter paper model generates unique millimeter paper textures, serving as backgrounds in scenes for renderings of leaves.

Our method is implemented in the fragment shader by procedurally modeling a texture of standard millimeter paper using sine functions. Given a local space position of a fragment (x, y) on the millimeter paper, the color intensity is determined by the following function:

$$C(x, y, \phi) = A \cdot \sin(B \cdot x + \phi) + D\,, \qquad (1)$$

where A is the amplitude of the stripes, which controls their intensity variation, B is their frequency, which determines the distance between them, ϕ is the phase shift which offsets them horizontally, and D is the baseline color intensity.

Further shader effects include hue, contrast, brightness, and saturation changes to emulate various paper conditions. Subsequently, we blend this texture with multiple layers of noise. We use three types of noise: gradient G, Voronoi V, and simple S. A selection of rendering results created by different parameter value configurations is shown in Fig. 2.

Composition of noise layers is given by a weighted sum:

$$L(x, y) = w_G \cdot G(x, y) + w_V \cdot V(x, y) + w_S \cdot S(x, y)\,,$$

where w are the weights determining the noise strength.

3.2. Procedural Leaf Model

Our algorithm for generating leaf models procedurally incorporates several computation stages to simulate leaf morphology. The initial shape is defined on the CPU using Unity's animation curve, which is a parametric piecewise polynomial curve defined by a set of control points that interpolate values to form a smooth transition.

Random perturbations are applied to the control points' positions to introduce variations, reflecting the inherent diversity found in leaf shapes. This randomness is mathematically expressed by adding a noise function N to the control point positions P, where the new position $P' = P + N$.

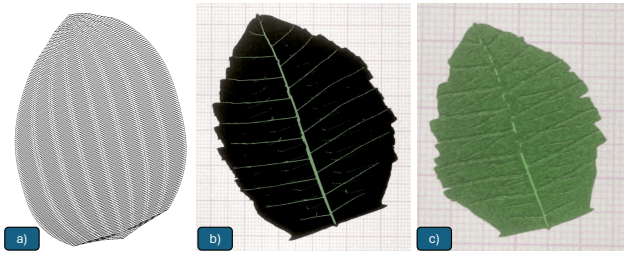

Figure 3. Procedural leaf model generation: The shape is defined by a parametric curve using Unity's Animation Curve (a), which is then textured, including vein pattern development (b), and stochastic elements and surface details are added via shader effects (c).

For vein patterning, the algorithm follows the method outlined by Goldman et al. [7], employing a turtle graphics system. This system is formalized as a series of commands defining the turtle's movement, where the turtle's path through space traces the veins - generated in screen space as a procedural texture. We add a stochastic element by incorporating Brownian Motion $Bm(t)$ for jittering the turtle's movement to create wavy vein lines and randomness in the branching angles θ_{veins} for the vein paths.

Texture generation extends to creating a height map H for depth representation, where H is modified by a function of the turtle's path and its corresponding vein thickness. The height map is used to obtain normals for the normal mapping within the fragment shader to produce detailed leaf surface textures (examples shown in Fig. 3).

Vertex displacements in the mesh are introduced using the Voronoi noise function V within the vertex shader to simulate the leaf's surface undulations. The fragment shader then employs a blend of procedural noises and a photorealistic leaf texture. Initially, we define two primary color textures, C_1 and C_2, representing different aspects of leaf coloration and patterning. The final leaf texture T is the result of blending these color textures, modulated by gradient noise $G(x, y)$. Additionally, detailed features are incorporated: veins (denoted as V_e, holes (denoted as H_o), and other textural elements such as spots and edge irregularities, which mimic natural imperfections in leaf morphology.

4. Implementation

Rendering Each leaf undergoes four separate rendering passes to capture varied appearances (examples shown in

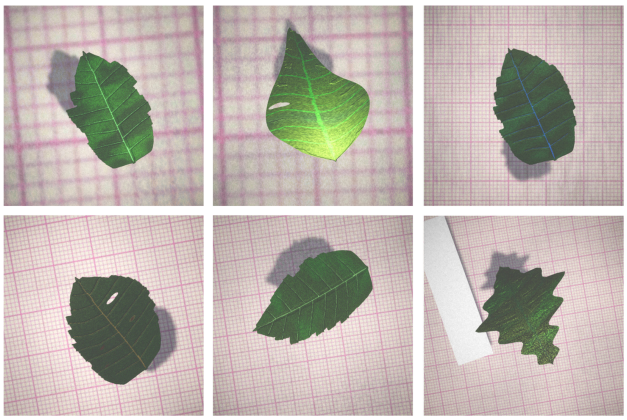

Figure 4. Diverse final renderings from the procedural leaf generation pipeline. This collection illustrates the variation achieved in leaf appearance through our procedural model parameters. Each rendering captures different lighting conditions, shadow effects, and background scaling.

Fig. 4). In each pass, we adjust shader parameters for shadow rendering, including strength (s), position ($\vec{p}$), and size (sz, referring to shadow size), to simulate various lighting environments. These parameters are governed by a shadow function $S(s, \vec{p}, sz)$, which is implemented using the shadow mapping algorithm.

Background Scaling The size of the millimeter paper in the scene is adjusted via a scaling factor γ, maintaining a consistent camera perspective across all renderings. The scaled background is denoted as $M' = \gamma M$.

Scene Composition Additional elements, like paper fragments and glass, are included to enhance realism. Their transformations in the scene are handled through a combination of randomized translations (T_{xy}), rotations (R_θ), and scalings (S_{xy}).

4.1. Data Preparation

An important aspect of our dataset is the generation of semantic masks and the precise computation of leaf area size labels. These elements are essential for various applications, including morphological analysis and ML model training.

We initiate another pass of the graphics pipeline for each leaf rendering with a uni-colored leaf mesh on a black background to obtain a semantic mask. The computation of the leaf area size is a direct application of the procedural parameters used in our millimeter paper model. Given that the paper model's grid is generated with known dimensions, and the scaling factor (γ) used in rendering is also known, we can accurately compute the area of each leaf by summing the areas of all triangles comprising the leaf's surface mesh.

Figure 5. Three instances of ControlNet-generated images where the region of the inpainted leaf in the mask deviates significantly from the region defined by the procedurally generated mask. Such data points are automatically filtered out in LAESI.

4.2. Integration of ControlNet Inpainting

Following the rendering of an initial 100,000 annotated synthetic images along with their masks, we use these images to obtain realistic 3D annotated counterparts. To achieve this objective, a pre-trained ControlNet network by Zhang et al. [26] is utilized.

In our implementation, the Stable Diffusion model [15] serves as the backbone, with a U-Net structure including an encoder, a middle block, and a skip-connected decoder.

The model uses down-sampling and up-sampling convolutions, ResNet blocks, and Vision Transformer layers for feature extraction. Textual inputs are encoded using a CLIP model, which is necessary for the generation process toward specific text-described attributes [12].

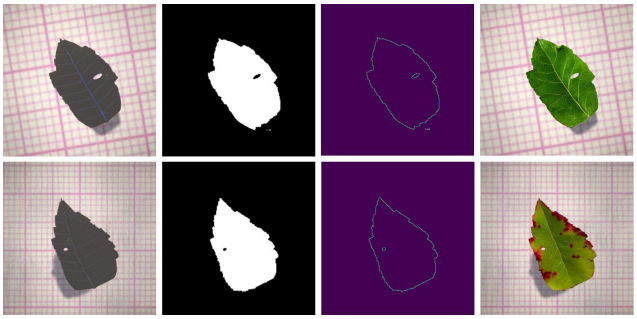

Figure 6. Example of inpainted results using ControlNet for semantic mask inpainting. In the lower row, the inpainting resulted with the addition of disease features, which were not described with the procedural model and would have been very challenging to simulate procedurally.

ControlNet provides a range of trained networks with diverse image-based conditions to regulate the diffusion models. These conditions include edges generated by various methods, depth, and normal maps, human poses, semantic segmentation, user sketches, etc. In our experiments, we employed the Canny-edge detection method [2]. Specifically, to generate realistic images from each synthetic image, we create an image using the Canny edges of our synthetic masks as inputs. The text prompt input consists of the specific phrases

Figure 7. Two pairs of synthetic (left) and real (right) images selected from the 100 highest cosine similarity scores from the Rendering 2 dataset and below from the ControlNet+Filtering dataset.

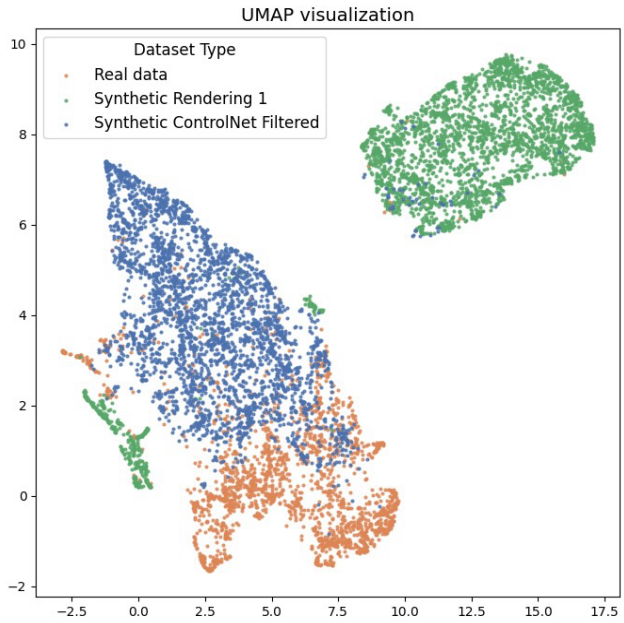

Figure 8. UMAP visualization of the ResNet50 (CLIP ViT-B/32) embeddings of real data (orange) and two different synthetic image sets (Rendering 1 - green, ControlNet + Filtering - blue). The lack of separation in the feature space between blue and orange dots suggests that the synthetic images for the ControlNet + Filtering dataset contain semantically more similar features compared to the Rendering 1 dataset.

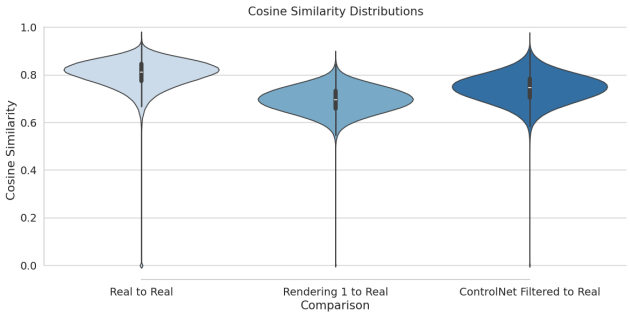

Figure 9. Violin plots of cosine similarity scores for datasets used in Fig. 8. ControlNet Filtered image distribution has overall higher cosine similarity scores compared to the Rendering 1 dataset. The distributions are obtained from images which have overall similar features compared to real ones as indicated by high scores.

"oak leaf on millimeter paper" and "beech leaf on millimeter paper", which provide input to ControlNet to inpaint the desired attributes of the synthetic leaf images. By incorporating this textual information, the inpainting process better matches the characteristics of oak or beech leaves. We then replace the background with our procedurally generated one. Note that we cannot use the mm paper background from the AI-generated image, or the leaf surface area annotations would become inconsistent.

Post-generation with ControlNet, the synthetic data undergoes a filtering process to eliminate images inconsistent with the associated annotations. This dataset refinement utilizes a MobileNet-based semantic segmentation model, trained on the 'synthetic rendering 2' dataset, comprising 5,000 synthetic and 1,700 real images (see Tab.1). Synthetic images, where the predicted mask deviates by more than 15% from the ground truth, such as in instances shown in Fig. 5, are filtered out of the dataset. This step removes outliers that we encountered during the ControlNet inpainting step in the frequency of 15-20%.

5. Validation

We apply synthetic data to train leaf area size prediction and semantic segmentation of real leaves on mm paper. This vision task is made deceptively difficult by using different sensors, camera extrinsic, and other image artifacts, such as reflections, notes, and objects appearing in the photographs in a real research environment. The application of rule-based leaf area prediction methods usually relies on either highly controlled environments [13] or specific reference objects included for scaling purposes [4], which makes these methods impractical for typical research compilation efforts.

5.1. Network Model Training

We used MobileNet V3 [9] to predict leaf area size through regression. We selected MobileNet V3, which has been de-

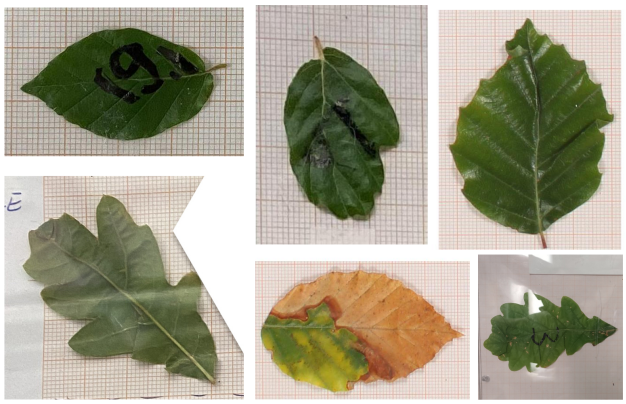

Figure 10. Photographs of beech and oak leaves on millimeter paper taken at the TUM School of Life Sciences. They were used to create a baseline for training a leaf surface area prediction and semantic segmentation network models and for the validation dataset.

signed for mobile applications running on hand-held devices to facilitate deploying our solution in remote locations.

5.2. Hyperparameter Optimization

Several hyperparameters were systematically varied during the experiments, including (1) Architecture variations (MobileNet V3 Large and Small). (2) Data augmentation techniques (brightness, contrast, hue, saturation, flip, 90^o rotation, random rotation, noise). Due to lack of identification of optimal hyperparameter configuration in the scientific literature, we employed a large hyperparameter space exploration by conducting 1,425 experiments to identify the most performant models through crowd sourcing (hyperparameter value configurations included in supplementary material).

Optimal hyperparameters were identified based on validation dataset performance. The best settings were MobileNet V3 Large with ImageNet transfer learning (pre-trained with ImageNet-1k weights), RMSprop optimizer, an initial learning rate of $1e^{-3}$ with Piecewise Constant Decay down to $1e^{-9}$ and augmentations including brightness, contrast, hue, saturation, flip, 90^o rotation, and Poisson noise.

5.3. Validation Experiments

We conducted six training experiments:

1. *Rule-based Baseline*: 30 real data points with uniformly-sized red squares for calibration purposes in the images. We calculated performance metrics with the "Easy Leaf Area" rule-based leaf area prediction method [4] in contrast to other experiments which use MobileNet V3.
2. *Real Data Baseline*: Training with 1,7K real, annotated leaf data points (examples shown in Fig. 10).
3. *Synthetic Rendering 1*: Training combined 1,7K real data points with 5K synthetic data points.
4. *Synthetic Rendering 2*: Training combined 1,7K real data points with 5K synthetic data points using improved parameter value configurations based on results obtained with previous dataset. We changed parameter values for the mm paper model to remove noisy paper effects by narrowing parameter value ranges of noise functions.
5. *Synthetic ControlNet*: Training combined 1,700 real data points with 5K, 6K, 7K, 8K, 9K, and 10K synthetic data points, using ControlNet for mask inpainting.
6. *Synthetic ControlNet + Filtering*: Training used ControlNet and a filtering process 5K, 6K, 7K, 8K, 9K, and 10K synthetic data points, mixed with 1,7K real data points.

Each experiment utilized the same validation and test datasets comprising 250 real annotated photographs. The validation data has been harvested from climate chamber experiments conducted at the TUMmesa ecotron and at the Technical University in Munich (TUM) in the years 2021 and 2022. Real annotations were obtained empirically and with the LICOR LI-3100C Area Meter.

5.4. Evaluation Metrics

The primary metric for evaluation is the mean relative error (MRE) of the leaf area size between predicted and ground truth values. For the semantic segmentation metric we employed the mean intersection over union (mIoU) of the ground truth to predicted masks, as well as the relative error in total count of mask pixels - called mask pixel error (MPE). Furthermore, we employ cosine similarity scores and the UMAP [11] dimensionality reduction method to quantify the similarity of synthetic and real images (Figs. 7- 9).

The outcomes of our synthetic training experiments, summarized in Table 1, reveal significant differences in model performance based on the type and amount of synthetic data used in training network models. The Real Data baseline experiment achieved an MRE of above 12.5% in validation and test sets. Rendering 1+2 experiments with mixed-in synthetic data demonstrated a slight improvement in validation MRE. However, the most significant advancements were observed in the Synthetic ControlNet and Synthetic ControlNet+Filtering experiments. Here, the inclusion of 10K synthetic data points with filtering substantially reduced the MRE to as low as 6.1% in validation and 6.2% in tests (Fig. 11). Interestingly, increasing the amount of data points results in a proportionally greater improvement in MRE for the ControlNet+Filtering compared to the other experiments. Furthermore, while the MRE for leaf area prediction improved for inpainted data (with and without filtering) the performance as measured by mIoU and MPE for the semantic segmentation model decreased for the ControlNet experiment (0.09 MPE) compared to the non-inpainted experiments (0.07 MPE) but increased for the ControlNet+Filtering experiment (0.05 MPE, Tab. 1). This shows the usefulness of filtering after the inpainting step improving significantly performance across the two vision tasks.

Table 1. Performance comparison of training experiments with real baseline (1,7K) and 5K synthetic training data, and a rule-based baseline.

Experiment	**Validation MRE** (%)	**Test MRE** (%)	**mIoU** (%)	**Mask Pixel Error** (%)
Rule-based Baseline	38.3	38.3	-	-
Real Data Baseline	12.5	12.9	0.79	0.08
Rendering 1	12.0	11.0	0.81	0.07
Rendering 2	10.8	11.3	0.82	0.07
ControlNet	8.4	10.0	0.8	0.09
ControlNet + Filtering	8.5	9.2	0.83	0.05

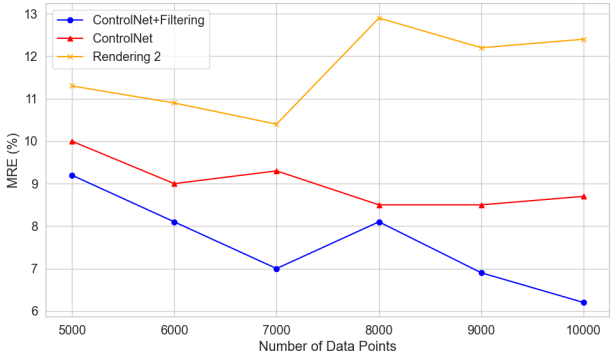

Figure 11. Validation data loss curves for experiments for training with datasets ranging from 5K to 10K data points. Red curve indicates results obtained with raw ControlNet inpainted data. blue curve with filtered data, and orange without inpainting (Rendering 2). While the addition of more data significantly improves the MRE on leaf area size prediction on inpainted data, there is no improvement for raw synthetic data.

6. Discussion and Conclusion

This study presents a fully automatic approach for generating synthetic data for leaf area prediction by utilizing procedural models and the MobileNet V3 neural network architecture. While standard computer vision methods for leaf area prediction (e.g., [4]) can work well in highly controlled contexts, we found that typical photographs compiled from wet lab experiments contain unexpected complexity making such methods challenging to use in practice. This includes notes, reflections, variable sensor intrinsics and extrinsics, pieces of paper, and other objects present in the images (see Fig. 6). Further, our results indicate that the inpainting of synthetic data with ControlNet is a feasible way for improving performance on a specialized vision task, and that the addition of filtering into the generative pipeline based on annotation consistency significantly improves overall performance. Specifically, our results indicate that the synthetic data before inpainting likely lacks sufficient feature variability to benefit training models on increasing quantities of datapoints (Fig. 11, orange curve), while the non-filtered inpainted data decreases model performance in the semantic segmentation task by introducing inconsistencies to annotations (Tab. 1, mIoU and MPE scores). Surprisingly, this means that the training on erroneous annotations in the ControlNet experiment still leads to better overall MRE in leaf area size prediction. Overall, our analyses indicate that the best domain adaptation for our procedurally generated synthetic data can be achieved via filtered inpainting which reduces annotation inconsistencies while preserving the AI-generated features. These findings are similar to those of Fei et al. [6] who show that using a semantic constraint loss in the training of GAN-based methods can help maintaining annotation consistency for improved model performance.

Through a series of experiments, we evaluated various hyperparameter value configurations and datasets, leading to an optimal setup that achieved a test loss of 6.2% in our final experiment, which is at least on par with the human annotation error and significantly outperforms the best real baseline. This led to the adoption of this model by TUM biology researchers in further experiments, making manual annotation redundant. The manual annotation of over 2,000 leaf images is a costly and laborious undertaking, prolonging the compilation of empirical results. Our results prove the value of synthetic data in training deep learning models for a specific research application, strengthening the claims of the usefulness of synthetic data made in other research work (e.g., [5, 10, 18, 25]).

Furthermore, our approach, which combines procedural generation with generative AI methods, holds promise for various other applications in botanical research and agriculture, particularly in remote sensing and precision agriculture. As future work, we would like to address the estimation of other growth parameters, such as shoot internode length or root biomass. Our approach can be extended to other domains, allowing fully automatic synthetic data generation for machine learning.

The project was funded by grant LWF-klifW021 from the Bavarian State Ministry for Food, Agriculture and Forestry (StMELF) to PA and FA. We thank Rudolf Harpaintner, Nooshin Mansouri, Barbara Brunschweiger and Patrick Mößmer for leaf images and/or measurements.

Figure 12. A selection of synthetic images generated with LAESI. These images are part of the ControlNet + Filtering subset.

References

[1] Dafni Anagnostopoulou, George Retsinas, Niki Efthymiou, Panagiotis Filntisis, and Petros Maragos. A realistic synthetic mushroom scenes dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6282–6289, 2023. 1, 2

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 8(6):679–698, 1986. 4

[3] Norishige Chiba, Ken Ohshida, Kazunobu Muraoka, and Nobuji Saito. Visual simulation of leaf arrangement and autumn colours. *The Journal of Visualization and Computer Animation*, 7(2):79–93, 1996. 2

[4] Hsien Ming Easlon and Arnold J. Bloom. Easy leaf area: Automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in Plant Sciences*, 2(7): 1400033, 2014. 5, 6, 7

[5] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training ... for now, 2023. 7

[6] Z. Fei, A. Olenskyj, B. N. Bailey, and M. Earles. Enlisting 3d crop models and gans for more data efficient and generalizable fruit detection. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1269–1277, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 7

[7] Ronald Goldman, Scott Schaefer, and Tao Ju. Turtle geometry in computer graphics and computer-aided design. *Computer-Aided Design*, 36:1471–1482, 2004. 3

[8] Sung Min Hong, Bruce Simpson, and Gladimir VG Baranoski. Interactive venation-based leaf shape modeling. *Computer Animation and Virtual Worlds*, 16(3-4):415–427, 2005. 2

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[10] Jonathan Klein, Rebekah E. Waller, Sören Pirk, Wojtek Pałubicki, Mark Tester, and Dominik L. Michels. Synthetic data at scale: A paradigm to efficiently leverage machine learning in agriculture. *Available at SSRN 4314564*, 2023. 7

[11] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 4

[13] Hadisseh Rahimikhoob, Mojtaba Delshad, and Romina Habibi. Leaf area estimation in lettuce: Comparison of artificial intelligence-based methods with image analysis technique. *Measurement*, 222:113636, 2023. 5

[14] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. *arXiv preprint arXiv:2306.09310*, 2023. CVPR 2023. 2

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4

[16] Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, and Przemyslaw Prusinkiewicz. Modeling and visualization of leaf venation patterns. In *ACM SIGGRAPH 2005 Papers*, pages 702–711. 2005. 2

[17] Adam Runions, Miltos Tsiantis, and Przemyslaw Prusinkiewicz. A common developmental program can produce diverse leaf shapes. *New Phytologist*, 216(2): 401–418, 2017. Special Issue: Plant developmental evolution. 2

[18] Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion, 2023. 7

[19] Tao Sun, Matthias Segu, Jan Postels, Yi Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: A synthetic driving dataset for continuous multi-task domain adaptation. In *CVPR*, 2022. 2

[20] Jordan Ubbens, Mikolaj Cieslak, Przemyslaw Prusinkiewicz, and Ian Stavness. The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant methods*, 14:1–10, 2018. 2

[21] Iris R Wang, Justin WL Wan, and Gladimir VG Baranoski. Physically-based simulation of plant leaf growth. *Computer Animation and Virtual Worlds*, 15(3-4):237–244, 2004. 2

[22] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. In *British Machine Vision Conference (BMVC) 2018, CVPPP Workshop*, 2018. arXiv:1807.10931 [cs.CV]. 1

[23] Qingan Yan, Jiahui Zheng, Sebastien Reding, Siyan Li, and Ivaylo Doytchinov. Crossloc: Scalable aerial localization assisted by multimodal synthetic data. *arXiv preprint arXiv:2112.09081*, 2021. 2

[24] Zongxin Yang, Ding Liu, Chen Wang, Jianchao Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *CVPR*, 2022. 2

[25] Zhuoran Yu, Chenchen Zhu, Sean Culatana, Raghuraman Krishnamoorthi, Fanyi Xiao, and Yong Jae Lee. Diversify, don't fine-tune: Scaling up visual recognition training with synthetic images, 2023. 7

[26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2, 4

[27] Y. Zhang, H. Ling, K. Gao, J. Yin, J.F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2