This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Plant Root Occlusion Inpainting with Generative Adversarial Network

Hao Song*
has236@mail.usask.ca

Karim Panjvani†
karim.panjvani@gifs.ca

Zhigang Liu†
zhigang.liu@gifs.ca

Huzaifa Amar†
huzaifa.amar@gifs.ca

Leon Kochian†
leon.kochian@gifs.ca;

Shengjian Ye‡
shengjian.ye@nrc-cnrc.gc.ca

Xuan Yang†
xuan.yang@nrc-cnrc.gc.ca

J. Allan Feurtado†
allan.feurtado@nrc-cnrc.gc.ca

Krunal Chavda*
krunal.chavda@usask.ca

Karina Angela Chimbo Huatatoca§
karina.chimbo@usask.ca

Mark Eramian*
eramian@cs.usask.ca

Abstract

Three-dimensional (3D) image analysis represents the state-of-the art for phenotyping in the fields of biology and plant science including studies of root system architecture. A widely used approach for capturing root architecture in 3D involves growth of roots in hydroponic media and capture of optical camera views via a stepper-motor-based rotation system. However, the introduction of structures to support 3D root growth system leads to significant occlusion of the roots during image acquisition, thereby causing the complexity and introducing inaccuracy of subsequent operations such as 3D modeling and root traits calculation. Instead of using a traditional manual sketching methods, this project proposes an automatic root gaps detection and inpainting method based on a Generative Adversarial Networks (GAN). The model was trained and evaluated using two distinct maize datasets, both of which were enriched with manually annotated segmentation and inpainting labels. The quantitative analysis of the inpainting results demonstrated variation in the performance of the GAN model. However, promising outcomes were observed with certain instances achieving Intersection of Union (IoU) and Dice Similarity Coefficient (DSC) values surpassing 0.9 with specific images or patches exhibiting lower accuracy and reproducibly. Despite this variability, the overall model performance maintained an average range of 0.8-0.9. Our GAN model presents a robust, effective and automatic solution for inpainting plant root gaps, leading to improved accuracy within the phenotyping pipeline. Moreover, the model demonstrates a great generality for inpainting other root system of species or cultivars beyond those encountered during training. The performance of the model exhibits superiority when confronted with less intricate root structures, but it produces less accurate results when confronted with complex root systems with large gaps or high root density.

1. Introduction

1.1. Motivation

The term *phenotype* refers to an organism's observable characteristics through its development including the morphological, biochemical or physiological properties. Plant phenotyping has become one of the essential topics in the context of biology, agriculture and plant science. Root system architecture (RSA), the shape and spatial arrangement of the root system, is one of the most significant parts support the plant's above-ground structure, and are responsible for uptake, storage and transportation of water and nutrients [8].

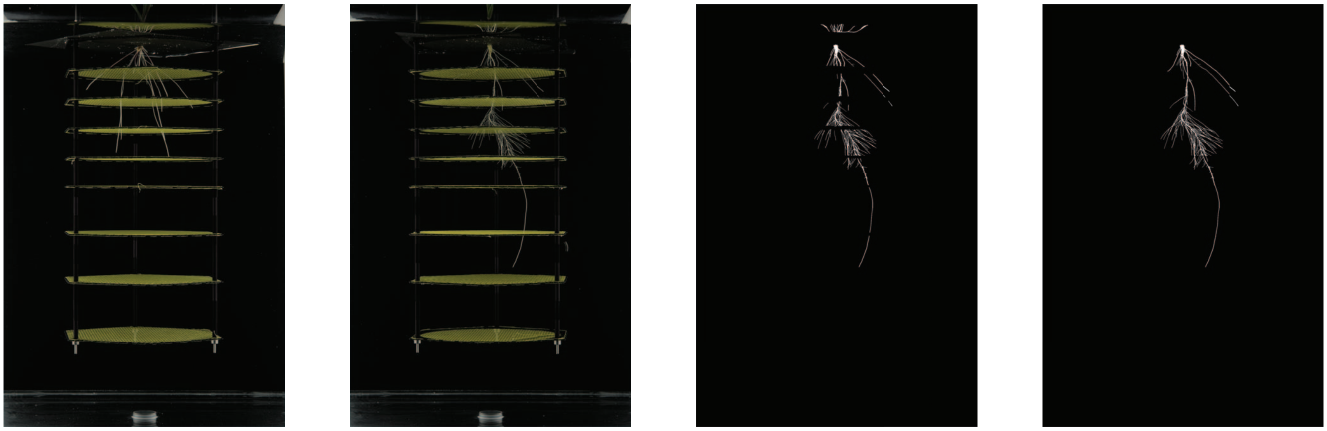
Non-invasive techniques such as hydroponic root imaging offers numerous advantages including rapid image acquisition, high-resolution, and premium visual quality. The introduction of supporting structure during root system development better reflects their natural growth environment in soil, and allows the root system to better maintain its spatial orientation during image acquisition. An example of such additional supporting structures can be observed in the work of Clark [4] and Piñeros et al. [8], wherein tower-

[*] Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada

[†] Global Institute For Food Security, 421 Downey Road, Saskatoon, SK, Canada

[‡] Aquatic and Crop Resource Development, National Research Council Canada, 110 Gymnasium Place, Saskatoon, SK, Canada

[§] Department of Plant Sciences, University of Saskatchewan, Saskatoon, SK, Canada



(a) Sample image 1

(b) Sample image 2

(c) Segmentation label

(d) Inpainting label

Figure 1: Sample images from training and testing datasets. (a) and (b) are sample hydroponic maize root images from the dataset. (c) is the segmentation result of (b) with manual annotation. (d) is the inpainting ground truth label of (b)

like structure was developed by weaving fishing net around hexagonal disks and employing plastic posts to support the multiple disks to aid in root support and stability. However, when capturing images, supporting structure occlude certain parts of the root system. As illustrated in Figure 1a, 1b and 1c this occlusion results in multiple horizontal and vertical gaps that fragment the root structure. Consequently, these gaps can introduce inaccuracies in the 3D modeling process, thereby compromising the precision of phenotype measurements. Manually sketching the missing root segments is a highly complex and time-consuming task. Hence, automating this process has became as a important objective in the advancement of fully automated, high-throughput Root System Architecture (RSA) analysis pipelines.

1.2. Deep Inpainting

In the realm of image processing, inpainting was widely used as a prevalent technique for the restoration and completion of missing data. This methodology entails predicting the absent data by leveraging contextual information from the surrounding regions such as color, texture, and line patterns. However, the intricate and non-repetitive structural characteristics of roots creates challenges for accurately predicting the contents over the gaps based on the surrounding areas alone. Consequently, traditional inpainting methods fail to predict missing root data.

Pathak et al. introduced an unsupervised adversarial network named Context Encoder [7]. This network introduced an encoder-decoder architecture where the encoder compresses a corrupted image into a compact representation, while the decoder reconstructs the data into a complete format. The optimization of the network utilizes mean square error (MSE) in combination with an adversarial loss. It should be noted that Context Encoder assumes prior knowledge of the missing region's location, which necessitates user input. In 2017, Sasaki et al. [10] proposed a deep inpainting approach that can identify and predict gaps in an image without the need for a mask indicating the missing regions. Yu et al. [13] took the approach of using random gap masks during training, demonstrating the model's capability to handle inpainting for various mask shapes.

Beyond the Context Encoder approach, Chen et al. [2] developed a deep learning model to predict random root gaps. They added gaps at the random locations in synthetic root images. Then they trained a convolutional neural network (CNN) model with an encoder and decoder to predict the root pixels. This methodology introduced an innovative approach for inpainting minor root gaps without requiring user input; however, it exhibits limitations in effectively detecting large occluded regions, as encountered within the hydroponic dataset referenced earlier.

1.3. Deep Inpainting based on GAN

A root patch that obtains a high evaluation score may yield a lower full-image evaluation score due to the limited information provided by the patch-level in assessing the completeness of the entire root. Thus, it's necessary to evaluate the model performance at both the patch-level and full-image-level. In order to address this, Chen's work was further improved in 2019 by incorporating local and global discriminators, enabling the model evaluation in both patch-level and full-image-level [3]. The inclusion of a global discriminator significantly enhanced the overall connectivity of the entire root structure, hence forms a Generative Adversarial Network (GAN).

This project builds upon, modify and extends Chen's GAN network, adding the capability to automatically identify and restore multiple larger and wider root gaps in the binary segmentation of root images that arise due to occlusion by supporting structures. The architecture of the network comprises three key components: an inpainting generator, a local discriminator, and a global discriminator. The inpainting generator is designed based on a U-net-like CNN. The local discriminator assesses the training process at the patch-level, while the global discriminator evaluates the training at a holistic full-image level. The model was trained with hydroponically-grown maize and pea roots, and subsequently, it was tested on two different datasets. During the testing phase, the model demonstrates remarkable performance in automatically and accurately filling horizontal gaps through inpainting, effectively restoring the completeness and connectivity of root structure. However, the current model has difficulty predicting area of root characterized by high density and overlapping. The proposed model presents a robust method for automatically inpainting missing root segments, offering potential enhancements for subsequent steps in the 3D reconstruction pipeline, such as 3D modeling, skeletonization, and root traits extraction.

2. Methodology

The model architecture depicted in Figure 2 comprises an inpainting generator, a local discriminator, and a global discriminator. The inpainting generator adopts a U-net-like CNN structure that is responsible for executing the inpainting task. In the evaluation process, the local discriminator compares the predicted patches with the corresponding ground truth labels to determine the veracity of the predictions. Additionally, the predicted patches are reassembled into a complete image, which is then evaluated against the full-image ground truth label for a comprehensive assessment of the inpainting performance at a global level.

In the training stage, the annotated segmentation labels serve as the input for training, while the inpainting labels are employed as the ground truth. At the preprocessing step, the model trainer first detects the locations and widths of gaps present within the image. Subsequently, the module proceeds to extract patches of the root along the identified gap regions with appropriate overlapping. Although the inpainting model is designed to accommodate a fixed patch size, the patching process leverages the flexibility of dynamic patch sizes by image re-scaling.

When training starts, the segmentation patches are then fed into the inpainting generator, which generates inpainted versions of the patches. Subsequently, a comparison between the segmentation patches and the corresponding inpainting ground truth patches takes place within the local discriminator. To evaluate the over-all inpainting performance, the predicted patches are stitched together to form a complete image, which is then compared against the full-image inpainting label in the global discriminator. The generator loss, local discriminator score, and global discriminator loss are combined in a linear manner to provide feedback to the generator model.

Amount	Unit	Neutrient
1	mM	$Ca\,(NO_3)_2 \cdot 4H_2O$
1	mM	NH_4NO_3
1	mM	KCl
0.85	mM	$MgSO_4 \cdot 7H_2O$
0.25	mM	$NH_4H_2PO_4$
77	μM	Fe-EDTA
25	μM	H_3BO_3
0.8	μM	$Na_2MoO_4 \cdot 2H_2O$
0.6	μM	$CuSO_4 \cdot 5H_2O$
9	μM	$MnCl_2 \cdot 4H_2O$
2	μM	$ZnSO_4 \cdot 7H_2O$

Table 1: Full strength nutrient solution table

During the testing phase, the segmentation root patches are fed into the thoroughly trained inpainting generator. The generator produces a set of patch-level probability maps as output, wherein pixels with higher probabilities indicate a greater likelihood of being root pixels. Subsequently, the predicted patches are combined to form complete root labels. Following this, the resulting labels undergo a series of post-processing steps, including edge anti-aliasing, dilation, and other relevant techniques.

2.1. Experiment Settings and Dataset Acquisition

Maize seeds were surface-sterilized with 8% (v/v) sodium hypochlorite for 20 min, rinsed with 18 Mega-ohm water (Ω), and germinated at 27°C on moist germination paper (Anchor Paper, St. Paul, MN, USA) in darkness. Four days after germination, seedlings with similar growth vigor were transferred to 300 L polypropylene tubs and grown hydroponically using a specially designed and constructed the three-dimensional root system architecture of plant growth system, which holds the seedling and root system on the mesh tower, as described by Piñeros et al.[9]. The full strength nutrient solution content can be found in table 1.

The pH of the solution was adjusted to 5.7 ± 0.1 with NaOH or HCl as required and was continuously aerated and replaced every three days. The seedlings were cultured in a growth chamber with 16 h of light at 27°C, 8 h dark at 22°C, a photosynthetic photon-flux density of around 350μmolm^{-2} s^{-1} at canopy height, and 50% relative humidity.

The acquisition of plant root images occurred at specific time points, namely on days 5, 7, and 10 following seedling emergence. Each day's dataset comprised 100 images per sample, captured using an optically corrected digital imag-

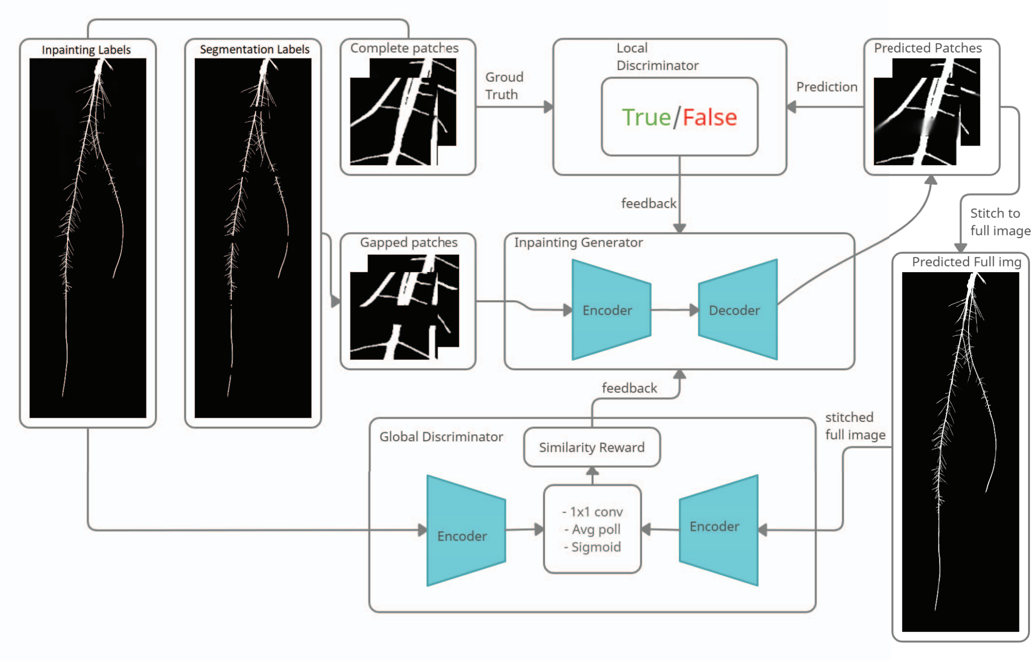


Figure 2: The GAN architecture and data processing pipeline.

ing system, and possessed a resolution of 5520×8288 pixels. The images were captured from a range of viewing angles, encompassing the full 360° rotation alone Y-axis, with an angular increment of 3.45°. It is noteworthy that the testing dataset encompassed diverse maize subspecies in contrast to the training datasets to ensure the randomization of the experiment.

The training dataset comprises 1,200 images captured from four pea samples and ten maize samples. Conversely, the testing dataset comprises 62 images sourced from the same datasets as the training set, alongside an additional 62 images obtained from a distinct dataset involving various maize species. The datasets underwent manual annotation by two annotators with distinct academic backgrounds using a ground-truth mark-up tool provided by Seidenthal [11]. Each image was assigned two labels, a segmentation label and an inpainting label. During the training phase, images have the potential to be divided into a variable number of patches ranging from 5 to 25, depending on the growth stage and size of the root.

Importantly, it should be noted that the training and testing methodology described herein is not limited to the specific plant root species of maize and pea. Rather, this approach can be readily applied to various other plant root species.

2.2. Model Architecture

2.2.1 Root gaps detection and patching

The occurrence of root gaps resulting from occlusion can be categorized into two types: horizontal gaps and vertical gaps. Horizontal gaps manifest as a consequence of occlusion caused by the presence of yellow mesh disks, as depicted in Figure 1a and 1b. Conversely, vertical gaps arise solely when a black plastic rod is rotated between the roots and the camera. The specific location and size of these gaps can vary, depending on factors such as the root size and image resolution. Moreover, the width of the gaps varies due to the perspective view. Notably, at any given time, only one vertical gap is present.

To identify the presence of root gaps, the row histogram was computed for each segmented root image. The identification of the root gap locations was subsequently accomplished by detecting the local minima within the histogram. Additionally, the widths of the wave troughs at these local minima were utilized as the approximate gap size. By leveraging the information pertaining to the location and width of each gap, patches were generated with overlapping approach to minimize the boundary effect. Notably, the patch sizes were dynamically determined based on the respective gap sizes, ensuring that the patches adequately encapsulated the full gaps. At last, all patches were resized into desired patch size before passing to generators.

2.2.2 Inpainting Generator

The inpainting generator consists of an encoder and a decoder, forming a fully convolutional network with a U-net-like structure. During the encoding phase, a patch is downsampled using a convolutional layer with a stride of 2. Consequently, patches with higher resolution are compressed into a latent space characterized by smaller dimensions. In the decoding process, the decoder module reconstructs and up-samples the latent code to generate a complete root patch with desired size (128 × 128 or 256 × 256). Notably, skip connections, reminiscent of those found in a U-net architecture, are incorporated to preserve information during the down-sampling steps, facilitating improved gradient flow, as in [3]. The output layer is comprised of a softmax activation function, yielding a probability map. Higher probabilities in the output patch signify a greater likelihood of the corresponding pixel being a root pixel.

This project retained the Cross-Entropy (CE) loss for validating the generator. The CE loss is obtained by comparing the probability map with the original patch (ground truth), which maximizes the likelihood of the training data distribution.

2.2.3 Local Discriminator

The local discriminator is responsible for evaluating the output patches. Mean square error, and mean absolute error are known to cause blurry issues when used in image generation problems [5, 7]. Also, Chen et al. argue that CE loss can cause inpainting to fail due to the multiple solutions to a gap [3]. When multiple predictions to a gap are averaged together, the generator has a higher chance of making a mistake inpainting the gap. Thus, a local discriminator network is used for learning data-driven loss, which will be added to the generator CE loss and improve the learning process.

The local generator focuses on the high-frequency parts since the overall low-frequency feature was represented by CE loss. Chen et al. [3] suggest using a Markovian discriminator to classify a sub-region of the patch as real or fake. The discriminator was applied convolutionally through the entire patch, and the average of all outputs is considered a final score of the patch. In addition, the discriminator uses least square loss instead of CE because it performs more stably and can produce better results. Also, a spectral normalization (SN) layer is added after each convolutional layer for constraining the Lipschitz constant of the classification function learned by the discriminator.

2.2.4 Global Discriminator

Evaluating the generator only in patch-level is not sufficient because plant root is a highly complex structure. High-quality prediction patches may still result low connectivity and completeness in full-image view. Thus, Chen et al. suggest adding another discriminator to validate the generator from the full-image view. The global validation can improve the generator performance and ensure a good result in both patch-level and global-level.

Since thresholding on the probability map and image concatenation are non-differentiable processes, backpropagation is unachievable in this step. Thus, traditional validation approaches are not suitable for this global discriminator. In this case, Chen et al. suggest using policy gradient [12] instead of CE loss. The generator is considered a policy network governed by the generator weight. It inpaints the gapped patches into a probability map, where the probabilities represent the possibility of a pixel being a root pixel. The probability map can be binarized by some standard image processing techniques such as thresholding or Bernolli sampling. Then the binarized prediction patches are stitched back into a complete root image, which we call the full-image prediction. The global discriminator encodes the full-image prediction and the original full-image ground truth into two latent spaces of size 512. Then, a sigmoid activation is applied on the dot product of two latent spaces and produces a full-image similarity score.

At last, the three loss functions will be combined linearly to get the final loss to feeds the generator. This process can be formulated as:

$$\mathcal{L}_G = L_{ce} + \lambda_1 L_{Local} + \lambda_2 L_{Global} \tag{1}$$

Where L_{ce} is the generator CE loss, L_{Local} is the loss from local discriminator, L_{Global} is the loss from global discriminator. Real constants λ_1 and λ_2 weight the influence of each loss.

2.3. Evaluation Metrics

In this project, we retained the common evaluation metrics used in segmentation problems. The binary ground truth images are annotated by scientists from both biology and computer science backgrounds. The quantitative evaluation metrics used are Intersection-Over-Union (IoU) and Dice Similarity Coefficient (DSC). IoU is also known as the Jaccard Index, one of the most commonly used metrics in semantic segmentation. The IoU is expressed as in Equation 2, where P is the set of pixels which are marked as foreground in the prediction image, and G is the set of pixels which are marked as foreground in the ground truth image. In addition, this paper evaluate the DSC index on the prediction. DSC can be obtained using Equation 3, with the same definition of P and G.

$$IoU(P,G) = \frac{|P \cap G|}{|P \cup G|} = \frac{|P \cap G|}{|P| + |G| - |P \cap G|} \tag{2}$$

$$DSC(P,G) = \frac{2|P \cap G|}{|P| + |G|} \tag{3}$$

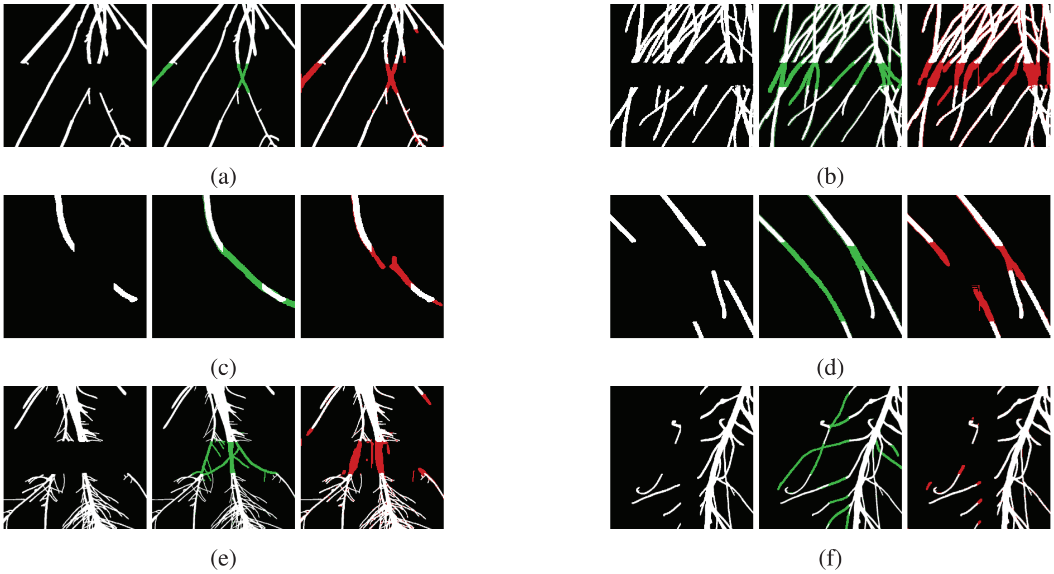

(a) (b) (c) (d) (e) (f)

Figure 3: Patch-level testing results, each sub-figure comprises a side-by-side combination of the original patch, a ground turth labelled by green color, and the corresponding inpainting results labelled by red color.

3. Experiments

The model was implemented in Anaconda python 3.6 [1] with Pytorch 1.0.1 [6]. The training was executed on a server with three NVIDIA GeForce 1080 graphic cards, which takes about 120 hours of execution time. The testing on a single image takes less than 5 seconds. The program can be executed on more GPUs if available.

3.1. Training

All training and testing images were stored in the PNG image format, which offers the advantage of lossless compression. The training dataset was randomly divided into training and validation sets, maintaining a ratio of 7 to 3. For the training processes, the patch size was set to 128×128 and a training batch size of 32 was selected, which can efficiently accommodate the server's graphic card memory size. In cases where a root sample consisted of fewer than 32 patches, appropriate repetition was applied. During the training process, the generator and discriminators employed the Adam optimizer. The model was trained for a maximum of 40 epochs, with training ceasing if the generator loss failed to improve for five consecutive epochs. In addition, the parameters for the Adam optimizer were set as follows: the learning rate was configured as 0.0002, the weight decay was set to 0, and the betas were specified as [0.5, 0.999].

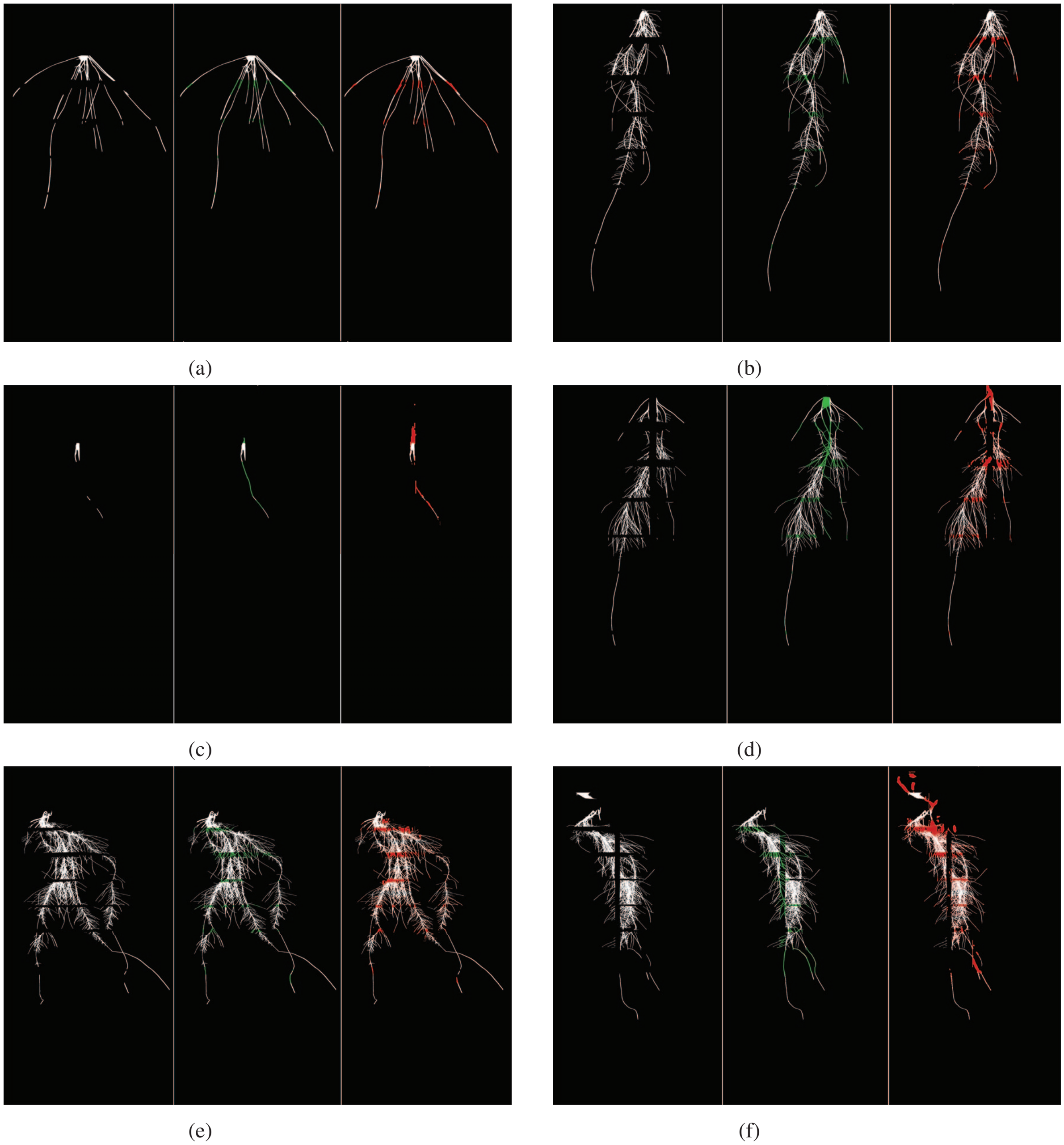
3.2. Testing

The GAN model was comprehensively tested on two distinct datasets, encompassing analyses at both the patch-level and full-image-level. The first testing dataset comprised 63 roots segmentation images , including two pea cultivars and 10 maize cultivars. Conversely, the second dataset encompassed 34 images of a single maize plant, captured from diverse view angles. Unlike dataset 1, dataset 2 encompasses crops in later growth stages, characterized by densely packed and overlapping root structures.

The evaluation of the testing images employed the root gap detection and patching techniques previously explained in Section 2.2.1. The testing images first undergo a patching process along the identified gaps using an overlapping method to minimize boundary effects. Subsequently, the resulting patches are fed into the trained GAN model for inpainting. Finally, the inpainted patches are reassembled to reconstruct the complete root structure at its original size. Subsequently, the evaluation metrics IoU and DSC (outlined in Section 2.3) will be computed at both the full-image and patches-level for further analysis and assessment.

4. Testing Results

In this section, we present the comprehensive testing results from both quantitative and qualitative perspectives. The evaluation of the testing results consists two distinct components: patch-level evaluation and full-image evaluation. A detailed statistical analysis of both datasets can be found in Table 2. Moreover, sample outputs for the patch-level and full-image evaluations are visually depicted in Figure 3 and Figure 4.



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

Figure 4: full-image testing results: (a), (b), (c) and (d) are sample test results from dataset 1 (same species as training datasets, (e) and (f) are sample test result from dataset 2. Each sub-figure comprises a side-by-side combination of the original patch, a ground turth labelled by green color, and the corresponding inpainting results labelled by red color.)

4.1. Patch-level Test Results

At patch-level, the IoU and DSC are ranged from 0.02 to 1, which means the model failed at some specific patches but generate perfect result on some others. The median value, located at near 0.8, indicates that it falls towards the higher end of the distribution. The mean values suggests that the dataset is moderately centered around this average. The variance of less than 0.02 indicates a relatively narrow spread of data points around the mean, reflecting a high level of consistency in the results.

Based on the visual inspection of sample images, it can be inferred that the model demonstrates satisfactory performance when dealing with uncomplicated root structures with less splitting and overlapping, as shown in Figure 3a and 3b. However, the model may encounter challenges

Dataset	Statistic	Min	Max	Median	Mean	Variance
1	IoU patch size	0.02	1	0.82	0.80	0.0203
	IoU full-image	0.31	0.94	0.82	0.81	0.0110
	DSC patch size	0.03	1	0.9	0.88	0.0114
	DSC full-image	0.47	0.97	0.9	0.89	0.0059
2	IoU patch size	0.02	1	0.82	0.80	0.0203
	IoU full-image	0.62	0.8	0.69	0.69	0.0018
	DSC patch size	0.03	1	0.9	0.88	0.0114
	DSC full-image	0.76	0.89	0.81	0.81	0.0009

Table 2: Quantitative Evaluation Statistics

when faced with large gaps, as well as instances where gaps occur at points of root splitting or in regions of high density, as shown in Figure 3c, 3d, 3e, and 3f.

4.2. Full-image Test Results

Significant evidence indicates a notable disparity in the model's performance between dataset 1 and dataset 2. The IoU and DSC values for dataset 1 ranged from 0.3 to 0.97. The median values for dataset 1 consistently fall within the range of 0.82 to 0.9. The mean IoU values for dataset 1 range from 0.81 to 0.88, indicating a relatively consistent and satisfactory performance. In contrast, dataset 2 demonstrates a substantial decline in median IoU to 0.69, suggesting a predominance of failures in accurately inpainting occlusion.

Figure 4a and Figure 4b present two samples of successful inpainting results. In Figure 4a, an IoU value of 0.88 and a DSC value of 0.93 were achieved. Similarly, Figure 4b attained an IoU of 0.87 and a DSC of 0.93. Notably, the model demonstrated its ability to effectively fill in the gaps in the root structures with minimal imperfections. Nevertheless, Figure 4c shows an instance where the model encountered difficulties in accurately inpainting a simple root structure with large gap. Furthermore, in Figure 4d, the model faced challenges in effectively addressing the vertical gaps where splitting and overlapping occur most. The IoU values for Figure 4c and Figure 4d are 0.46 and 0.67, respectively. These outcomes exemplify instances where the model struggled to achieve satisfactory results when the root gaps are larger than the patch scale and when root gaps covering root branching and overlapping, highlighting its limitations in handling certain types of root structures and complex inpainting scenarios.

Figure 4e and Figure 4f display two sample outputs obtained from dataset 2. In Figure 4e, an IoU value of 0.78 and a DSC value of 0.88 were achieved. Conversely, Figure 4f attained an IoU of 0.62 and a DSC of 0.76. The testing results on dataset 2 highlight the model's limitations in handling root structures with massive density and overlapping. The model tends to either utilize solid blocks of labels when filling gaps or ignore the gaps if inpainting cannot be done. These observations underscore the need for further improvements in the model's ability to handle complex root structures with densely intertwined regions.

5. Conclusion

This project introduces a Generative Adversarial Network that automatically apply detection and inpainting on root gaps occurring within a hydroponic imaging system for 3D modeling of the root system architecture (RSA). The proposed model leverages the power of deep learning techniques, eliminating the need for manual intervention. It demonstrated versatility by being trainable on various RSA in different species, and testable on different growth stages and viewing angles. Although the model achieves a high level of accuracy on simple root structures with lower density and minimal overlap, it encounters challenges when predicting over-scale gaps and highly dense root regions. This project has made a significant contribution to the development of a comprehensive 3D plant root processing, analysis, and phenotyping pipeline. By effectively restoring missing root data, recovering root connectivity, and enhancing the accuracy of 3D model reconstruction, this project exhibits considerable potential in assisting root structure architectures analysis for plant scientists and biologists.

6. Acknowledgment

This is to acknowledge all those whose invaluable contributions have materialized this endeavor. A significant acknowledgment is extended to the Operations Team at the Global Institute for Food Security (GIFS) for their indispensable aid and support in the operation of the imaging system. In particular, we deeply appreciate of the exceptional hardware and facility support from Zongyu Wang and Garret Wilson.

References

[1] Anaconda software distribution, 2020. 6

[2] Hao Chen, Mario Valerio Giuffrida, Sotirios A. Tsaftaris, and Peter Doerner. Root gap correction with a deep inpainting model. In *British Machine Vision Conference*, 2018. 2

[3] Hao Chen, Mario Valerio Giuffrida, Peter Doerner, and Sotirios A Tsaftaris. Adversarial large-scale root gap inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 5

[4] Randy T. Clark, Robert B. MacCurdy, Janelle K. Jung, Jon E. Shaff, Susan R. McCouch, Daniel J. Aneshansley, and Leon V. Kochian. Three-dimensional root phenotyping with a novel imaging and software platform. *Plant Physiology*, 156(2):455–465, 2011. 1

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 5

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[7] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2, 5

[8] Miguel A. Piñeros, Brandon G. Larson, Jon E. Shaff, David J. Schneider, Alexandre Xavier Falc ao, Lixing Yuan, Randy T. Clark, Eric J. Craft, Tyler W. Davis, Pierre-Luc Pradier, Nathanael M. Shaw, Ithipong Assaranurak, Susan R. McCouch, Craig Sturrock, Malcolm J. Bennett, and Leon V. Kochian. Evolving technologies for growing, imaging and analyzing 3d root system architecture of crop plants. *Journal of Integrative Plant Biology*, 58(3):230–241, March 2016. 1

[9] Miguel Piñeros, Brandon Larson, Jon Shaff, Dave Schneider, Alexandre Falcão, Lixing Yuan, Randy Clark, Eric Craft, Tyler Davis, Pierre-Luc Pradier, Nathanael Shaw, Ithipong Assaranurak, Susan Mccouch, Craig Sturrock, Malcolm Bennett, and Leon Kochian. Evolving technologies for growing, imaging, and analyzing 3d root system architecture of crop plants. *Journal of Integrative Plant Biology*, 58:230–241, 03 2016. 3

[10] Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Joint gap detection and inpainting of line drawings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5725–5733, 2017. 2

[11] Kyle Seidenthal. Friendly ground truth, 5 2021. https://github.com/p2irc/friendly_ground_truth. 4

[12] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000. 5

[13] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 2