Adversarial Large-scale Root Gap Inpainting

Hao Chen
University of Edinburgh
s1786991@ed.ac.uk

Mario Valerio Giuffrida
University of Edinburgh
v.giuffrida@ed.ac.uk

Peter Doerner
University of Edinburgh
Peter.Doerner@ed.ac.uk

Sotirios A. Tsaftaris
University of Edinburgh
The Alan Turing Institute
S.Tsaftaris@ed.ac.uk

Abstract

Root imaging of a growing plant in a non-invasive, affordable, and effective way remains challenging. One approach is to image roots by growing them in a rhizobox, a soil-filled transparent container, imaging them with digital cameras, and segmenting root from soil background. However, due to soil occlusion and the fact that digital imaging is a 2D projection of a 3D object, gaps are present on the segmentation masks, which may hinder the extraction of finely grained root system architecture (RSA) traits. Herein, we develop an image inpainting technique to recover gaps from disconnected root segments. We train a patch-based deep fully convolutional network using a supervised loss but also use adversarial mechanisms at patch and whole root level. We use Policy Gradient method, to endow the model with large-scale whole root view during training. We train our model using synthetic root data. In our experiments, we show that using adversarial mechanisms at local and whole-root level we obtain a 72% improvement in performance on recovering gaps of real chickpea data when using only patch-level supervision.

1. Introduction

The analysis of the root system architecture (RSA) of a plant is an important area of plant phenotyping (and breeding) research, as plant roots are responsible for acquiring water and nutrients from the soil (or surrounding environment). However, root phenotyping requires apparatuses for plant root visualization, which remains challenging due to soil opacity. Destructive root phenotyping (*shovelomics*), performed by digging out the roots from the soil, requires significant human resources and does not offer long-term observations of the same plant. Non-destructive root visualization methods, such as MRI imaging [35] or X-ray tomography [24] are expensive and require specialized equipment. Growing roots in visible, soil-based containers, such as rhizoboxes [4], and imaging them in greenhouses using digital cameras (refer to Figure 1(a)), is a more affordable and robust method. RSA parameters could be estimated and analyzed on segmentation masks of root images, where roots are separated from the background (as in Figure 1(b)).

(a) (b) (c)

Figure 1. (a) A digital image of a root in rhizobox; (b) segmentation mask of root; (c) our inpainting result. Orange box emphasizes the gaps present on the root due to occlusion of soil.

However, in rhizoboxes the full RSA is not visible: root segments may contain discontinuities, which are present in the segmentation mask of a root, as shown in Figure 1(a) and (b). The presence of such gaps restricts the extraction of finely-grained RSA parameters, e.g. tip counts and number of secondary roots. Therefore, resolving gaps on segmentation masks is of significant value to extract more detailed information from the RSA. Human annotation would not be efficient enough, as it is tedious and error-prone. It is difficult for humans to decide which segments should be connected together, as a root may have thousands of tips and branches, and due to occlusion by soil. We argue that the problem of filling the gaps in root system architecture can be solved with the use of machine learning.

Plant phenotyping tasks have been recently approached with machine learning: e.g. plant segmentation [22], leaf segmentation [34, 39, 42] and counting [7, 10]. In several tasks, the performance with deep learning has been consid-

erable [38]. Herein, we propose a solution for root gap filling problem using deep convolutional neural networks.

The problem of recovering gaps in the RSA can be addressed with image inpainting. Image inpainting is a technique of a long history and has numerous applications, e.g. damaged paintings restoration and objects removal, which refers to modifying an image in an undetectable way. Study of inpainting algorithms started in [3] and became prosperous from [2] where inpainting of low computation by matching the corrupted image with nearest-neighbour patches within training data was proposed. More recently, deep learning based inpainting has been delivering both realistic and natural inpainting results [13, 27, 40].

Relating inpainting techniques with root phenotyping, the authors in [5] proposed a deep learning method to automatically fill in the gaps in chickpea root segmentation masks, which were obtained from imaging system. The model was trained on a synthetic dataset, as no ground truth labels exist for chickpea roots. Although promising results were shown in filling gaps on chickpea root, three main limitations still exist. Firstly, the employed Mean Absolute Error (MAE) reconstruction loss produces blurry results, especially when dealing with complex patterns. Secondly, domain mismatch between synthetic root and chickpea root results in poor performance of the model on chickpea full images, as synthetic root differs from chickpea root in terms of scale, texture, and gap patterns. Thirdly, due to the relatively large dimension of synthetic root images which cannot be fitted in GPU memory during back-propagation, the authors used root patches to train their model. This introduced bias of semantic understanding about the root as the model does not have a global view of the RSA, which should contain only one fully connected component. Yet in root patches, several root segments caused by cropping may exist (see Figure 3(c) ground truth for example).

In this paper, we propose a model (Figure 2) to address these limitations. We cast the root inpainting task as a classification problem, where Cross Entropy loss is optimized and a categorical distribution is imposed on the model. We improve the capability of our model to inpaint complex gap patterns and produce sharper results by adding a discriminator that looks at local patches [11, 14]. To circumvent limitations due to GPU memory and endow the model with the ability to obtain feedback from local (patch-based) choices based on performance on the whole root (global-based), we adopt an additional global discriminator that interacts with our inpainting generator through Policy Gradient [36]. Finally, to bridge the domain mismatch between synthetic root and chickpea root data, we randomly augment synthetic root during training to make them more visually similar to chickpea roots. We evaluate our model using image fidelity scores (Mean Square Error) and proxy metrics. All the results show that our model has better generalisation capacity and could produce higher quality inpainting results.

Our contributions are summarized as follows:

- We use adversarial learning to train the network to recover missing root segments, including one local discriminator and one global discriminator.
- We adopt Policy Gradient method in reinforcement learning to include a global view of the whole root during training, avoiding memory issues due to large image dimensions and allowing non-differentiable process to be used during training.
- We demonstrate our model could perform better in inpainting chickpea root segmentation masks without actually training on real data.

The rest of the paper is organized as follows. Section 2 discusses related work; Section 3 details our proposed method and Section 4 shows our experimental results. Finally, Section 5 concludes the manuscript.

2. Related Works

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) proposed in [11] and its variants show a great breakthrough in capturing multi-modal data distributions and they have been employed to solve a multitude of computer visions tasks, such as image generation [28, 29], image super-resolution [16], and image inpainting [13, 27].

The general framework of GANs consists of a generator network G and a discriminator network D competing in a minimax game [11]. G is a generative network that optimizes Jensen-Shannon divergence between empirical data distribution p_{data} and generated data distribution p_G. The generator tries to produce realistic-looking images w.r.t. a latent distribution p_z, while the discriminator aims to classify (and detect) between real and generated data.

Although GANs have achieved outstanding results, they can be unstable during training and suffer from mode collapse [11, 32]. To overcome these problems, several variants of GANs have been proposed, optimizing other objectives, such as Pearson χ^2 divergence in LSGAN [37], and Wasserstein distance in WGAN [1]. More general, any f-divergence could be used to optimize GANs [25]. We adopt least square loss [37] to train our model as it produces images of high quality and is more stable during training. To further stabilise training of our discriminator we use Spectral Normalization (SN) [23] on each convolutional layer, which regularizes the weights.

2.2. Inpainting with Deep Learning

In [27], the Context Encoder, an unsupervised adversarial network for inpainting, was proposed: the encoder compresses the corrupted image into a compact representation,

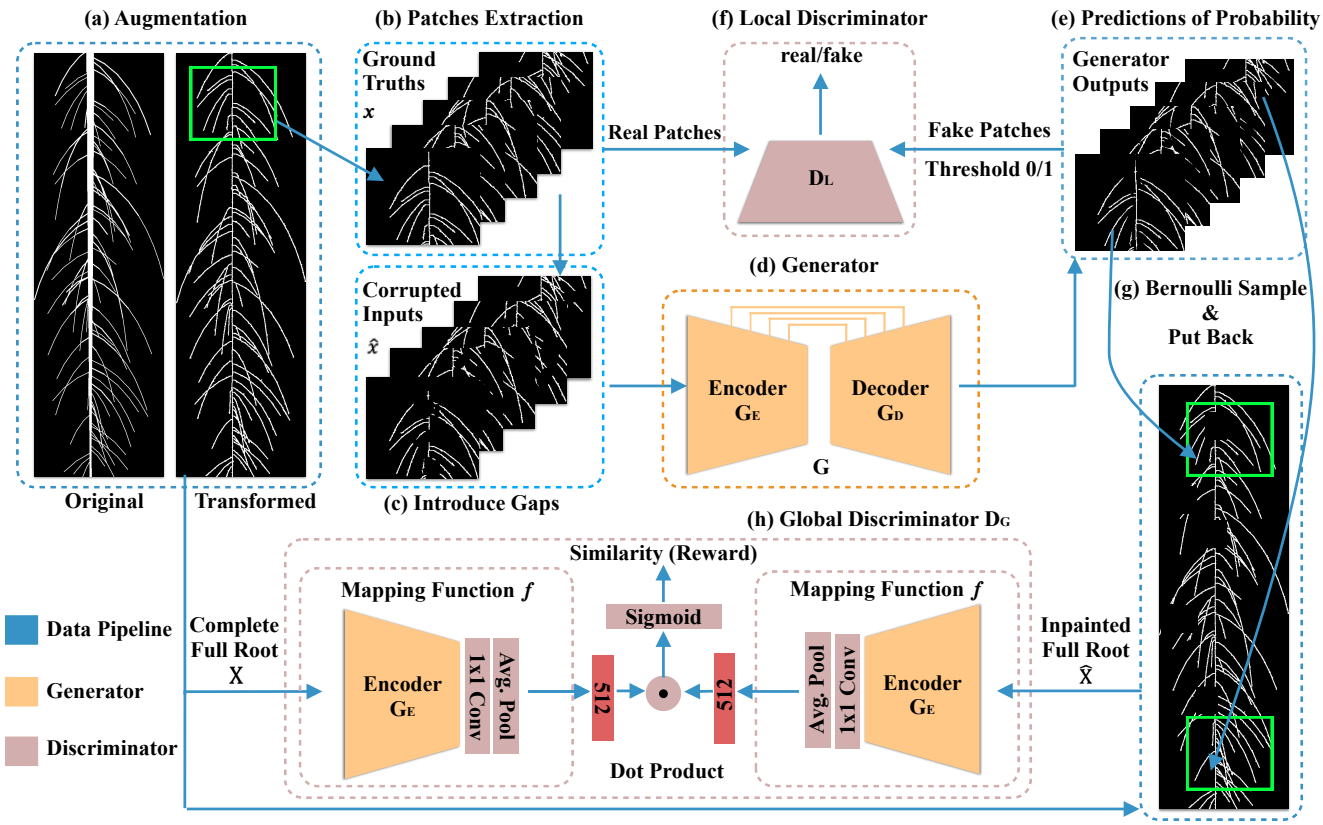

Figure 2. Model Architecture and Data Pipeline. (a) Augmentation is performed on a synthetic root. (b) Non-overlapping patches are extracted from the augmented root. (c) Random gaps introduced into patches to make corrupted patches input. (d) Generator network G that inpaints. (e) The prediction of the probability map obtained by G. (f) Local discriminator classifies inpainted patches and ground truth patches. (g) Bernoulli sampling is conducted on probability maps from G to obtain binary inpainted results, which are put back to the transformed root where they were extracted from to obtain an inpainted whole root. (h) Global discriminator computes the similarity score over the complete whole root and inpainted whole root.

whereas the decoder reconstructs a complete version of it. The network is optimized end-to-end with a Mean Square Error (MSE) supervised loss in combination with an adversarial loss to obtain sharper images. However, Context Encoder assumed that the region to be inpainted is given, which is not the case in our application. In [33], the authors demonstrated that line drawings have enough structure to allow the model to automatically detect and inpaint the gaps without the need for masks indicating the missing regions. In [40], random masks of gaps of any shape and at any position were used for training. Combining with a patch discriminator, their model could conduct free-form inpainting. While a discriminator that improves local inpainting quality could be used, the authors in [13] include an additional discriminator for global inpainting consistency.

Plant root has a similar thin-structure as line drawings; thus, may have enough structure to be inpainted automatically. Based on this, authors in [5] employed the fully convolutional encoder-decoder model defined in [33] to recover gaps from the RSA of chickpea plants. As ground truth inpainting labels for chickpea roots were not available, their model was trained using a synthetic root dataset [18]. Due to fully supervised training using Mean Absolute Error (MAE) loss, the model tends to average all possible outcomes together when making predictions, which results in blurry inpaintings and poor performance of the model when complex gaps are present. Inspired by [13], in this paper we add two discriminators into our system, one for inpainting quality and another one for global consistency of root, to alleviate the problems brought by MAE.

2.3. Policy Gradient

Policy Gradient (PG) is an approach broadly used in reinforcement learning [36], where a policy π (e.g. a neural network) predicts probabilities of action selection $\pi_\theta(a|s)$, given a representation of state s as input and its policy parameters (weights) θ. A value function $Q^\pi(s, a)$ finds the scalar reward r given the action a selected, and the weights are optimized according to maximum expectation of the reward. In this paper, we focus on REINFORCE, i.e. Monte-

Carlo Policy Gradient, for updating the weights [36]:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{PG}(\theta) &= \nabla_\theta \mathbb{E}[Q^\pi(s,a)\pi_\theta(a|s)] \\ &= \nabla_\theta \mathbb{E}[r \log \pi_\theta(a|s)]\end{aligned} \tag{1}$$

One can see that the policy network can learn without the requirement of the value function to be differentiable or backpropagation through the value function. The gradients are computed from logarithm of the policy output, which are scaled by corresponding scalar reward. PG has been used in many other fields other than reinforcement learning, especially in text generation [21, 30, 41], where directly optimizing metrics on sentence level is highly desirable. However, the process of sampling words from their probability vectors to form a sequence is non-differentiable. The authors in [6] used REINFORCE to sidestep this problem in adversarial training of image captioning, where a discriminator is trained to compute the similarity between textual descriptions generated from the generator and the given images. The similarity score was used as a reward when training the generator using PG.

Motivated by this work, we include an additional global discriminator in our system, who calculates the similarity score between inpainted whole root and complete whole root and feedback reward with generator through PG.

3. Methodology

We develop our model using a U-net like [31] fully convolutional model that follows an encoder-decoder structure as shown in Figure 2(d). We use Cross Entropy (CE) loss with softmax activation for the last layer of our inpainting network G to solve the root inpainting task as classification problem, as root segmentation masks contain binary values (either background or root), rather than a regression problem where continuous values are predicted.

As ground truth inpainting labels are not available in chickpea root dataset, we train our model to inpaint a synthetic root dataset [18]. We augment the synthetic root dataset to make them more chickpea-like. Gaps of random shape are introduced at random locations on the root to produce corrupted inputs and complete target training pairs.

Two discriminators are included in our model, one local discriminator D_L and one global discriminator D_G. D_L classifies real/fake at local patch level, i.e. 128×128, that is, the local discriminator informs to the generator the goodness of the produced inpaintings, such that the generator can improve the quality of the inpainted images. D_G computes a similarity score between the inpainted roots and ground truth ones. When updating the weights of the generator, D_G is functioning on whole roots level and feeds the information back to the generator through Policy Gradient [36]. The generator is thus encouraged to produce patch inpainting results that improve the completeness of the whole root.

Our pipeline is shown in Figure 2. In the following sections, we detail how we augment synthetic data for training, the design of our model, and the objectives we used.

3.1. Dataset and Augmentation

Dataset. Obtaining labeled data from real RSA is difficult, error-prone, and time-consuming. Inspired by [19, 8], we use publicly available synthetic root data that is artificially generated instead to train our model [18]. This synthetic dataset contains dicot and monocot roots of large resolution (see Figure 2(a)). Since the original images are saved in JPEG format, compression artifacts are removed as described in [5]. Directly training on such large full images would lead to GPU memory issues during backpropagation. Therefore, instead of using full images for training, we extract non-overlapping patches.

Augmentation. Although a model trained on this synthetic root dataset has been illustrated that it could also inpaint chickpea root images [5] , domain mismatch still exists between chickpea and synthetic root images in terms of root scale, root texture and gap patterns. Chickpea roots are characterized by high tortuosity, whereas synthetic roots are smoother and more regular [5]. Furthermore, gaps on chickpea root usually leave relatively random gap patterns (edges). In [5], the authors train the model by introducing structural square gaps, where the edges of gaps are mostly vertical, leading to overfitting of the model by only recognizing such edges. To bridge the domain mismatch, we augment the synthetic dataset. For a synthetic root image, we first skeletonize it to obtain a structure of the root, and then we randomly add noise onto the root skeleton edge to obtain serpentine textures. Dilation of random iteration is then applied to root skeleton to obtain various root thicknesses. This process makes the synthetic root similar to the real chickpea RSA. After the transformation, we extract patches of different sizes, i.e. from 128×128 to 384×384, which are re-scaled to 256×256 to capture the scale variability. We then introduce gaps of random shapes at random locations on the root to obtain training pairs, where gap masks are made according to the algorithms provided in [9, 40] that leave more random gap patterns.

3.2. Inpainting Generator

The generator network G is a fully convolutional network which has an encoder G_E and decoder G_D, with a bottleneck in between. In the encoding path, a corrupted root patch is downsampled using convolutional layers of stride 2, where the relatively high-resolution root patch, i.e. 256×256, is mapped into a latent space of smaller dimension. In the decoding path, the generator reconstructs a complete root patch using the latent code coming from the bottleneck. We use skip connections [31] to allow the sharing of features learned at shallow layers between encoder

and decoder to preserve information lost during down sampling. These skip connections also provide better gradient flow to the model. The last layer of the decoder is a convolutional layer producing a feature map with 2 channels. A softmax activation is then applied on the final feature map to obtain a probability map. The loss is computed between the probability map and training ground truth labels using cross-entropy (CE) to maximize the likelihood of training data distribution. The formulation of the CE objective is:

$$\mathcal{L}_{ce} = \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[\log(G(\hat{x}))] + \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[1 - \log(G(\hat{x}))] \quad (2)$$

where $\hat{x}$ is sampled from corrupted data distribution $p_{\hat{x}}$.

3.3. Local Discriminator

Mean Square Error (MSE) and Mean Absolute Error (MAE) reconstruction losses tend to produce blurry results in image generation problems [14, 27]. We argue that CE loss suffers from the same problem as it also fits single-modal distribution to the model, at which all possible outcomes are averaged together. This results in poor performance of the model to deal with complex gap patterns, as there are numerous possible inpainting solutions, the model does not know what to do and would leave it as a gap (refer to Figure 3 baseline model results).

To obtain sharper results, we use a local discriminator network D_L to learn a data-driven loss. Normally, a discriminator takes the inpainted image from the generator or a ground truth image as input, and learns to discriminate between these two. However, as overall low-frequency features could already be captured by the CE loss, we only need the discriminator to focus on local high-frequency details as CE tends to average them. Thus, we adopt a Markovian discriminator as in [14], which classifies a 128×128 patch in an image as real or fake. The discriminator is run convolutionally across an 256×256 patch, where all outcomes are averaged to obtain the final score for the patch. We replace the sigmoid cross entropy loss function in the vanilla GAN [11] with least square loss [20], as it has a more stable training dynamic and enables the model to produce higher quality results. We add Spectral Normalization (SN) [23] layer for each convolutional layer in our local discriminator, except for the last one. The SN layers constrain the weights of each convolutional layer, which shares a similar idea of WGAN [1], limiting the Lipschitz constant of the classification function learned by the discriminator. More specifically, the formulation of the local adversarial loss is:

$$\begin{aligned}\mathcal{L}_{LocalDadv} = \mathbb{E}_{x\sim p_x}[(1 - D_L(x))^2] \\ + \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[D_L(G(\hat{x}))^2],\end{aligned} \quad (3)$$

$$\mathcal{L}_{LocalGadv} = \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(1 - D_L(G(\hat{x}))^2], \quad (4)$$

where p_x is the empirical distribution of the training data.

3.4. Global Discriminator

Since the model is trained at a patch level, it has never seen the appearance of a complete whole root. When the model makes predictions on an input image, it may produce results of high inpainting quality but may lack full-root-level completeness (see Figure 3(a) for an example). Therefore, a global discriminator that can judge inpainting results on whole root level is desired, where good inpainting results should not only be consistent within a patch but also improve the completeness of the whole root. D_G is used at both patch, when updating its own weights, and at global level, when updating the weights of the generator G.

Due to memory issues introduced by training models on large images and the existence of a non-differentiable process during training, i.e. thresholding the probability map to obtain binary output and placing the resulting patch back into the full image, we adopt Policy Gradient [36]. We approximate the gradient for the generator from the global discriminator, where back-propagation on full images can be avoided, and where non-differentiable functions can be used in training process.

We view our generator as a policy network, governed by its weights. It takes corrupted root patches as input, and produces probability map patches, indicating the probability of each element being a root pixel. A Bernoulli sampling is then conducted on the probability maps to obtain binary inpainting results. Afterward, the inpainted patches are put back to the whole root according to locations representing where they come from to obtain an inpainted whole root. The global discriminator takes the inpainted whole root and original complete whole root as inputs, transforming them into a latent space of dimension 512 through a mapping function f. The dot product of these two latent features are computed and a sigmoid activation is employed on the results to obtain the similarity (reward) score of these two whole roots. We formulate this process as:

$$r(X, \hat{X}) = \sigma(f(X), f(\hat{X})), \quad (5)$$

where we use X to denote the whole root examples from which the patches used for training the generator were extracted. $\hat{X}$ represents inpainted whole root examples composed by the sampled probability maps.

To avoid back-propagation on full images while updating the global discriminator D_G, we still use patches to train it to minimize the similarity error. In contrast, the generator is trained with reward from D_G on whole root images. We train G and D_G with the cross entropy loss, as the last layer of D_G is a sigmoid function:

$$\begin{aligned}\mathcal{L}_{GlobalDadv} = \mathbb{E}_{x_1,x_2\sim p_x}[\log(r(f(x_1), f(x_2)))] \\ + \mathbb{E}_{\hat{x_1}\sim p_{\hat{x}}}[\log(1 - r(f(\hat{x_1}), f(x_2))],\end{aligned} \quad (6)$$

$$L_{GlobalGadv} = \mathbb{E}_{\hat{X}\sim p_{\hat{X}},\hat{x}\sim p_{\hat{x}}}[r(f(\hat{X}), f(X))\log(\hat{x})], \quad (7)$$

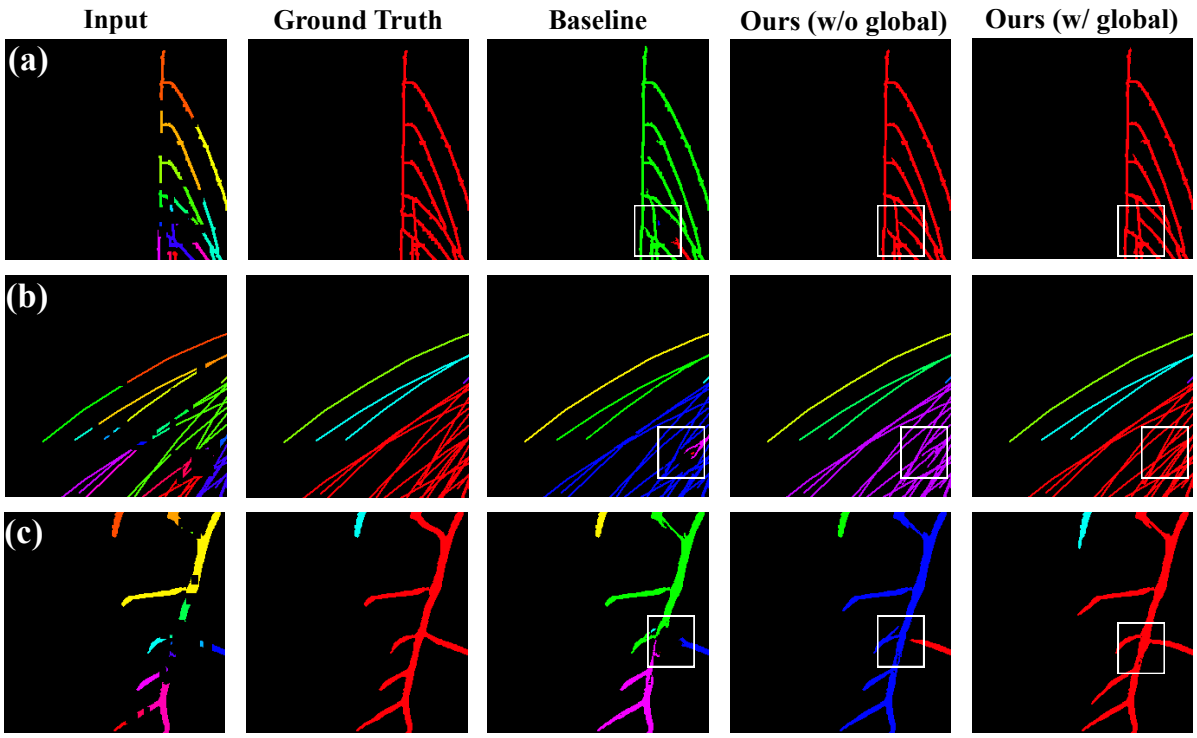

Figure 3. Qualitative results comparison on patch level. We use different colours to indicate different segments caused by gaps. (a) and (b) are synthetic root examples. (c) is a chickpea root example. A white box emphasizes local differences.

where both x_1 and x_2 are complete patches examples but extracted from different root and $\hat{x}_1$ are corresponding probability maps generated by generator network G.

Given an input image of size $H \times W$, we define the feature extraction function f of D_G as a combination of the encoder part of the generator G_E, an additional 1×1 convolutional layer that imposes a linear transformation of the features maps from G_E to $\frac{H}{4} \times \frac{W}{4} \times 512$, and an average pooling layer over each feature map which supports D_G to accept images of different dimensions (as shown in Figure 2(h)). For each image, the output from f is a vector of dimension 1×512 irrespective of the image's dimensions. We assume that the encoder of our generator could detect the gaps present on the root and there is no need to learn another network to conduct the same task. Noteworthy is that the encoder part of the generator G_E is updated twice within an training iteration, when updating the global discriminator weights and generator weights.

3.5. Total Training Objectives

The gradient for updating the generator comes from: (i) the supervised cross entropy inpainting loss; (2) patch discriminator loss; (3) policy gradient of the reward from global discriminator. The final training loss for the generator is:

$$\mathcal{L}_G = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{LocalGadv} + \lambda_2 \mathcal{L}_{GlobalGadv}, \quad (8)$$

where λ_1 and λ_2 balance the influence of each loss.

4. Experiments

Model Architectures. For the generator G, we adopt the same architecture that is detailed in [5], except the last convolutional layer which has 2 filters instead of 1. A softmax activation is applied to the feature maps produced by the last convolutional layer to obtain probability maps. For the local discriminator D_L, we use 5 consecutive convolutional layers with a stride of 2. Each convolutional layer has a kernel size of 5. The number of filters of the convolutional layers in D_L follows $[64, 128, 256, 256, 256]$. The global discriminator D_G consists of a mapping function f we defined in Section 3.4. D_G functions sequentially on 2 input roots to extract feature vectors of them, where the dot product of 2 feature vectors is calculated and sigmoid activation is applied to obtain similarity.

Training details. Our model is implemented in Pytorch [26]. Training data pairs are created from mixed dicot and monocot synthetic root images [5]. We optimize eq. (8) for the generator, eq. (3) for local discriminator D_L and eq. (6) for global discriminator D_G. We set $\lambda_1 = 4e^{-3}$ and $\lambda_2 = 2e^{-3}$ in our experiments. Adam [15] optimizer is adopted for both generator and discriminators with different learning rate [12], i.e. $2e^{-4}$ for generator and $4e^{-4}$ for discriminators. The model is trained for approximately 36 hours on a single NVIDIA TITAN Xp GPU, yet inference

	Metrics	MSE		# Pixel Diff.		# Connected Component Diff.	
	Models	Overall	Within Gaps	g.t. - input (before inpaint)	g.t. - pred. (after inpaint)	g.t. - input (before inpaint)	g.t. - prediction (after inpaint)
Synthetic	Baseline	0.0042± 0.003	0.8069± 0.167		257.4± 182.6		1.9± 3.5
	Ours (w/o global)	0.0027± 0.002	0.7845± 0.169	694.7± 388.2	190.9± 125.3	9.9± 4.8	0.9± 2.4
	Ours (w/ global)	**0.0025± 0.002**	**0.7289± 0.171**		**161.4± 113.8**		**0.3± 0.7**
Chickpea	Baseline	0.0035± 0.002	0.9136± 0.083		226.2±126.1		1.1±1.0
	Ours (w/o global)	0.0033± 0.002	0.8723± 0.084	1,171.5± 259.9	215.2± 122.8	8.3± 3.0	0.4± 0.5
	Ours (w/ global)	**0.0028± 0.002**	**0.8427± 0.073**		**185.7± 115.3**		**0.3± 0.5**

Table 1. Comparison of synthetic root patches ($N = 1500$) and chickpea patches ($N = 150$) using MSE loss, number of pixel difference and number of fully connected component difference. We compute the difference between ground truth and input, and ground truth and predictions. For chickpea patches, we visually selected a set that appears complete. We set $\lambda_2 = 2e^{-3}$ for including global discriminator.

on a 256×256 patch takes only a few milliseconds.

Evaluation setup. We evaluate our model and compare with the baseline in [5] with qualitative and quantitative results on synthetic patches, real chickpea patches, and chickpea whole root images. We test models on 1500 synthetic patches that are randomly selected from our augmented synthetic whole root images containing both dicot and monocot roots, and 150 chickpea patches, which were manually selected to avoid not including gaps, in Section 4.1. MSE loss over the full patch and MSE only within gaps are computed respectively to measure the image fidelity between inpainted results and ground truths. Proxy metrics, such as difference of the number of pixels and difference of the number of fully connected components are also used. Experiments about inpainting quality on 25 full real chickpea root images are conducted in Section 4.2. We use the number of fully connected components to evaluate the completeness of the inpainted results. We also use the root analysis software Root Image Analysis-J (RIA-J) [17] to measure several traits, i.e. the root length, tip counts and convex hull area, to show how they are 'recovered' by our models.

4.1. Patch-Level Results of Real and Synthetic Root

Here, we test our model on 1500 synthetic patches and 150 real chickpea patches. The synthetic patches are obtained from random selection from the synthetic roots in our test set, which contains dicot and monocot equally. The chickpea patches are chosen manually from full chickpea root to ensure they are as complete (and without gaps) as possible so that image fidelity scores and difference of proxy metrics could be computed.

We show a number of qualitative comparisons in Figure 3 between our model, where the global discriminator is turned off (i.e. $\lambda_2 = 0$) and a baseline model [5]. Root segments resulting from discontinuities are colored, where the less number of colors indicates a more complete root. From these results, one can observe that the baseline model has poor performance when dealing with complex gap patterns: it could not inpaint them as there are too many possible results and the baseline model does not know what to inpaint. By combining the baseline model with a local patch discriminator, there is an improvement already shown in terms of the number of fully connected components. A local discriminator helps the model to inpaint more accurately, yet it is not powerful enough, leaving some gaps. A global discriminator further boosts performance, by providing inpainting results that makes the root more complete.

These observations hold also quantitatively across a representative testing set as the comparisons in Table 1 shows. Image fidelity measurements such as MSE and pixel difference are computed. Lower MSE loss and lower pixel difference, indicate the model is inpainting more accurately, according to the ground truth. To show the capability of the models to recover root segments, we compute the difference of the number of fully connected components, which should be as close as to zero. We performed a paired t-test when comparing the results of our adversarial model and baseline model, where a highly significant difference is shown with a two-tailed p value < 0.0001 (highlighted as a bold font). The superiority of combining local and global adversarial losses is statistically evident, where our model could inpaint roots more precisely; pixel difference has been significantly decreased and the number of fully connected components are reduced to be under one (on average).

4.2. Complete Root-Level Results of Chickpea

It is more difficult to evaluate results of whole chickpea roots as no ground truth exists. However, we carefully select 25 real chickpea whole root images. We highlight that there is no ground truth on these chickpea whole root images and the model has never seen real (chickpea) images.

We display qualitative results in Figure 4. The baseline model could inpaint several gaps present; but shows worse performance compared to the proposed. Even using only a local discriminator, larger and more complex gaps can be inpainted compared to baseline.

On these 25 real images, we also assess improvement from a phenotypic trait preservation perspective as described previously in Section 4. We compute the relative improvement for each metric between the inpainted output

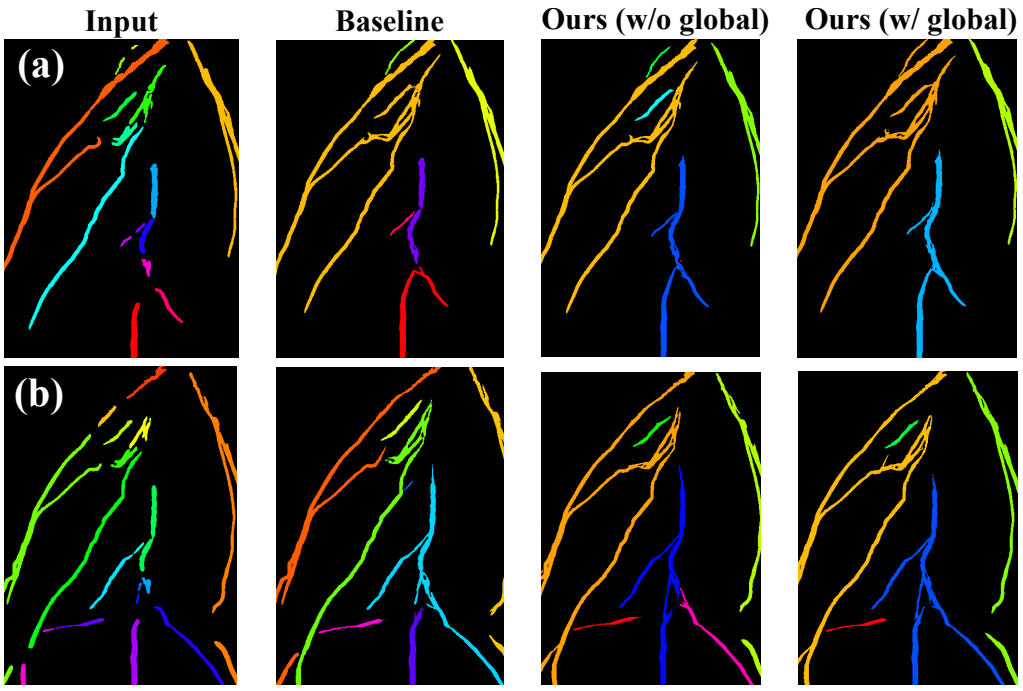

Figure 4. Qualitative results comparison on whole root level. We use different colours to indicate the different segments caused by gaps. (a) and (b) are examples of real chickpea roots inpainted with the method in [5] and two variants of the proposed method without and with global discriminator.

	Baseline	Ours (w/o global)	Ours (w/ global)
# Fully Conn. Comp. (%)	58.82±38.8	63.91±47.4	**68.88±55.5**
Root Length (%)	7.84±7.6	7.16±7.0	8.1±8.4
Tip Counts (%)	20.00±10.5	25.00±14.9	29.2±18.4
Convex Hull Area (%)	60.9±7.8	63.9±10.0	70.2±10.6

Table 2. Comparison of real chickpea whole root results ($N = 25$) in terms of relative improvement (after inpainting) on the number of fully connected components, root length, tip count, and convex hull of inpainted results compared to original corrupted ones.

and before inpainting, normalised by the metric value before inpainting. As shown in Table 2, our model can produce inpainting results that have lower number of fully connected components (an indication of being more complete), which is statistical significant over the results of baseline model, indicated by a paired t-test. The improvement on tip counts and convex hull area is considerable (albeit statistically not significant, probably due to small sample size). The original chickpea roots have many segments, resulting in inaccurate tip counts, which our model could 'repair' effectively. Also, inpainted results from our model have larger convex hull area, indicating more discontinuities shown in original chickpea root are reconnected together in inpainted ones.

5. Conclusion

We present an effective approach for filling gaps in the roots visualized via affordable plant root phenotyping systems. Our approach is trained in an adversarial way, with a local discriminator to encourage local high-quality inpainting results and a global discriminator to produce inpainting results with a global root view. We include global view of root through Policy Gradient [36], which is the first attempt of combining reinforcement learning techniques with large-scale image inpainting. Our results are comparable to state-of-the-art results on thin-structure inpainting.

Future works should focus on including real data during training, with a mechanism that involves some of the real data could help improve the results. Finally here we used proxy metrics to judge performance but ultimately human expert evaluation can judge whether our models improve root structure preservation.

Acknowledgements

This work was supported by the BBSRC grant BB/P023487/1 (http://chickpearoots.org) and also partially supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223, 2017. 2, 5

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009. 2

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM, 2000. 2

[4] Thibaut Bontpart, Cristobal Concha, Valerio Giuffrida, Ingrid Robertson, Kassahun Admkie, Tulu Degefu, Nigusie Girma, Kassahun Tesfaye, Teklehaimanot Haileselassie, Asnake Fikre, Masresha Fetene, Sotirios A. Tsaftaris, and Peter Doerner. Affordable and robust phenotyping framework to analyse root system architecture of soil-grown plants. *bioRxiv*, 2019. 1

[5] Hao Chen, Mario Valerio Giuffrida, Sotirios A. Tsaftaris, and Peter Doerner. Root gap correction with a deep inpainting model. In *British Machine Vision Conference (BMVC)*, page 325. BMVA Press, 2018. 2, 3, 4, 6, 7, 8

[6] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional GAN. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2989–2998, 2017. 4

[7] Andrei Dobrescu, Mario Valerio Giuffrida, and Sotirios A. Tsaftaris. Leveraging multiple datasets for deep leaf counting. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2072–2079, 2017. 1

[8] Clément Douarre, Richard Schielein, Carole Frindel, Stefan Gerth, and David Rousseau. Transfer learning from synthetic data applied to soil-root segmentation in x-ray tomography images. *J. Imaging*, 4:65, 2018. 4

[9] Emilien Dupont and Suhas Suresha. Probabilistic semantic inpainting with pixel constrained cnns. *arXiv preprint arXiv:1810.03728*, 2018. 4

[10] Mario Valerio Giuffrida, Peter Doerner, and Sotirios A. Tsaftaris. Phenodeep counter: a unified and versatile deep learning architecture for leaf counting. *The Plant Journal*, 96(4):880–890, 2018. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 2, 5

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 6626–6637. Curran Associates, Inc., 2017. 6

[13] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, July 2017. 2, 3

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 2, 5

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 6

[16] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 2

[17] Guillaume Lobet. guillaumelobet/Root-Image-Analysis-Pipeline- Evaluation: Release 2.0. https://doi.org/10.5281/zenodo.208499, Dec. 2016. 7

[18] Guillaume Lobet, Koevoets Iko T, Pierre Tocquin, Loïc Pagès, and Claire Périlleux. Library of simulated root images. https://doi.org/10.5281/zenodo.61739, Sept. 2016. 3, 4

[19] Guillaume Lobet, Iko T Koevoets, Manuel Noll, Patrick E Meyer, Pierre Tocquin, Claire Périlleux, et al. Using a structural root system model to evaluate and improve the accuracy of root image analysis pipelines. *Frontiers in plant science*, 8:447, 2017. 4

[20] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, Oct 2017. 5

[21] Luke Melas-Kyriazi, Alexander Rush, and George Han. Training for diversity in image paragraph captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 757–761, Brussels, Belgium, 2018. Association for Computational Linguistics. 4

[22] Massimo Minervini, Mohammed M. Abdelsamea, and Sotirios A. Tsaftaris. Image-based plant phenotyping with incremental learning and active contours. *Ecological Informatics*, 23:35–48, 2014. 1

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *2018 International Conference on Learning Representations (ICLR)*, 2018. 2, 5

[24] Sacha J Mooney, Tony P Pridmore, Jonathan Helliwell, and Malcolm J Bennett. Developing x-ray computed tomography to non-invasively image 3-d root systems architecture in soil. *Plant and soil*, 352(1-2):1–22, 2012. 1

[25] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 271–279. Curran Associates, Inc., 2016. 2

[26] 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. In *NIPS-W*, 2017. 6

[27] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016. 2, 5

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[29] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[30] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017. 4

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, pages 234–241. Springer, 2015. 4

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2234–2242. Curran Associates, Inc., 2016. 2

[33] Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Joint gap detection and inpainting of line drawings. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5768–5776, 2017. 3

[34] Hanno Scharr, Massimo Minervini, Andrew P French, Christian Klukas, David M. Kramer, Xiaoming Liu, Imanol Luengo, Jean-Michel Pape, Gerrit Polder, Danijela Vukadinovic, Xi Yin, and Sotirios A. Tsaftaris. Leaf segmentation in plant phenotyping: a collation study. *Machine Vision and Applications*, 27(4):585–606, may 2016. 1

[35] Hannes Schulz, Johannes A Postma, Dagmar van Dusschoten, Hanno Scharr, and Sven Behnke. Plant root system analysis from mri images. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*, pages 411–425. Springer, 2013. 1

[36] Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 1057–1063. MIT Press, 2000. 2, 3, 4, 5, 8

[37] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, June 1999. 2

[38] Sotirios A. Tsaftaris and Hanno Scharr. Sharing the right data right: A symbiosis with machine learning. *Trends in plant science*, 2019. 2

[39] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. In *British Machine Vision Conference (BMVC)*, 2018. 1

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2, 3, 4

[41] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *The Association for the Advancement of Artificial Intelligence (AAAI)*, pages 2852–2858, 2017. 4

[42] Yezi Zhu, Marc Aoun, Marcel Krijn, Joaquin Vanschoren, and High Tech Campus. Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. *British Machine Vision Conference (BMVC)*, 2018. 1