Novel data augmentation strategies to boost supervised segmentation of plant disease

Clément Douarre, Carlos F Crispim-Junior, Anthony Gelibert, Laure Tougne, David Rousseau

Version of Record: https://www.sciencedirect.com/science/article/pii/S0168169919304879
Manuscript_4836571f1d8a8c230fc422ab59f3375c

Novel data augmentation strategies to boost supervised segmentation of plant disease images

Clément Douarre[a,b,c,*], Carlos F. Crispim-Junior[a], Anthony Gelibert[c], Laure Tougne[a], David Rousseau[b]

[a] *Univ Lyon, Université Lyon 2, LIRIS, UMR CNRS 5205, F-69676, Lyon, France*
[b] *Laris, UMR INRA IRHS, Université d'Angers, 62 Avenue Notre Dame du Lac, 49000 Angers, France*
[c] *Carbon Bee, Rue du Commerce, ZA Les Plaines, 26320 St Marcel-Lès-Valence, France.*

Abstract

Annotation of images in supervised learning is notably costly and time-consuming. In order to reduce this cost, our objective was to generate images from a small dataset of annotated images, and then use those synthesized images to help the network's training process. In this article, we tackled for illustration with agricultural material the difficult segmentation task of apple scab on images of apple plant canopy by using convolutional neural networks. We devised two novel methods of generating data for this use case: one based on a plant canopy simulation and the other on Generative Adversarial Networks (GANs). As a result, we found that simulated data could provide an important increase in segmentation performance, up to a 17% increase of F1 score (a measure taking into account precision and recall), compared to segmenting with weights initialized on ImageNet. In this way, we managed to obtain, with small datasets, higher segmentation scores than the ones obtained with bigger datasets if using no such augmentations. Moreover, we left our annotated dataset of scab available for the plant science imaging community. The proposed method is of large applicability for plant diseases observed at a canopy scale.

1. Introduction

In computer vision applied to agriculture, machine learning techniques are currently progressing very rapidly (Minervini et al., 2015; Pound et al., 2017; Kamilaris and Prenafeta-Boldú, 2018). Two categories of methods are usually distinguished in machine learning depending on whether it is supervised or unsupervised. Supervised techniques require annotation, a task that is possibly time-consuming when dealing with images with complex structures, such as texture, or with large dataset of images, as in phenomics for plant sciences. This task is even more time-consuming now with the success of deep learning methods. These methods show great success (Goodfellow et al., 2016) on a variety of image processing tasks of seemingly unlimited complexity thanks to sufficiently large neural networks. However, as these networks are composed of many computational units, they typically require a great quantity of annotated data compared to other learning methods. Yet, in the case of plant disease detection, annotated datasets of infected plants are often very small. Often, the severity of a given infection is judged by the quantity of visible lesions: Hence, image segmentation *i.e.* pixelwise classification is needed, which requires pixelwise annotation, which is tedious and difficult.

Our goal was to obtain the best segmentation score possible using a neural network based segmentation and data simulation techniques with very small size of the datasets. We investigated the possibility to circumvent manual annotation by the generation of synthetic annotated data. Such generators fall in three broad categories:(i) Standard data augmentation, which has just started to be tested in plant sciences, (ii) computer graphics techniques of image rendering and (iii) generative adversarial networks (GANs) (Goodfellow et al., 2014) which is a new approach, not based on computer graphic model but on machine learning. These three kinds of generators have only recently been tested separately in plant sciences (Pawara et al., 2017; Ubbens et al., 2018; Giuffrida et al., 2017) mainly with images of single healthy plants. In this work, we implemented and compared an example of each of these categories on the same plant disease use case at the observation scale of the canopy.

We focus on apple scab which is an infection afflicting apple trees and in general trees of the *Malus* genus. This infection is caused by fungus *Venturia inaequalis*, which leads to brown lesions on the leaves and the fruits. It requires more than ten fungicide treatments per year to be controlled and can be considered as the most serious disease for apple plantations (Bowen et al., 2011). Quantifying the development of scab is of great importance for studying interactions of apple scab and apple trees, as well as for analyzing the evolution of pathogenicity in *Venturia inaequalis* populations and for breeding scab-resistant apple cultivars. In order to improve its treatment,

*Corresponding author
Email addresses: clement.douarre@liris.cnrs.fr (Clément Douarre), carlos.crispimjunior@univ-lyon2.fr (Carlos F. Crispim-Junior), anthony.gelibert@carbonbee.fr (Anthony Gelibert), laure.tougne@liris.cnrs.fr (Laure Tougne), david.rousseau@univ-angers.fr (David Rousseau)

early detection, *i.e.* before visible symptoms would be valuable. The life cycle of the fungus starts by a phase under the leaf's cuticle (invisible to the naked eye), before rupturing to the leaf's surface, at which point infection is visible but already very advanced (Oerke et al., 2011). However, the early stage is detectable by observation of other wavelengths than the visible spectrum. Indeed, in that stage, the fungus blocks the host leaf's water evacuation, which in turn modifies the temperature of infected zones. Infrared imagery, which provides information on temperature, is therefore suitable for early detection (Belin et al., 2013; Delalieux et al., 2009). The infrared domain has shown optimal contrast compared to RGB (Oerke et al., 2011; Chéné et al., 2012; Belin et al., 2013; Benoit et al., 2016). The detection of apple scab was implemented on individual plants at a very early stage of development where plants had a very limited amount of leaves (Oerke et al., 2011; Chéné et al., 2012). In a real agronomic and biological research environment, populations of trees are grown and the observation scale of plants is rather at canopy level.

This work is positioned along the following lines toward the closest related literature on data augmentation. Most simulation of plant images has been conducted by the implementation of biological models to generate 3D scenes. These simulations are then coupled with an artificial acquisition system in order to generate 2D images. Full plant simulation framework have been developed, enabling researchers to model a wide variety of crops and plants, along with many physical measurements on these models (Pradal et al., 2008). Some of these simulations were used to boost machine learning algorithms. For example simulator of root images were developed, which also generated annotation masks for a segmentation task (Benoit et al., 2014). Closer to our work, some simulators were designed to serve as dataset augmentation. In recent years, a weed simulator was designed to improve weed detection (Di Cicco et al., 2017), and high-quality 3D scenes of pepper plants were generated for a model of species identification (Barth et al., 2018). Some researchers focused, like this work, at the leaf level: 3D *Arabidopsis* leaves were generated in order to improve leaf segmentation (Ward et al., 2018). While these elaborate models generate highly realistic images, they often require many complex steps going through sometimes proprietary software, such as Blender. The model-based simulator presented here is deliberately simpler than the ones presented. It was simply based on a Python script and an open-source dataset, and worked in two dimensions. Also, it simulated scab disease, which is a finer level of detail than simulating whole species. Generative Adversarial Networks have now much improved from their debuts (Goodfellow et al., 2014). While the original GAN strived to generate images from a given true image distribution, the key concept of generating images for a given conditional distribution was then introduced (Mirza and Osindero, 2014). The condition the authors used was a class label: the conditional GAN, or cGAN, learned to generate images of specific digits from the MNIST dataset (instead of learning to generate all digits indifferently). This was shown to actually improve the visual quality of GAN results on that dataset compared to using no labels, and it also paved the way for a new range of applications for this kind of network. For exemple, one example of using such cGANs for data synthesis is generating different kinds of liver legion images in order to improve their classification (Frid-Adar et al., 2018). The cGAN's principle was generalized by providing images and strings as inputs instead of class labels (Isola et al., 2017). Images as inputs contain much more information than integer labels. Thus, the objective of such cGANs shifted from actual generation to domain transform, *e.g.* transforming maps to satellite images. Segmentation was also studied by (Isola et al., 2017), considering the image and its annotation as the two domains of interest. While results were quite visually impressive, such segmentation was not competitive with benchmarks of dedicated segmentation networks. The process of generating annotation from images (or vice-versa) has since been applied as a mean of data synthesis to augment datasets before performing segmentation with those dedicated networks. In plant sciences, a recent example concerns *Arabidopsis* leaf counting, where a cGAN was trained to generate plant images using the number of leaves as the label (Giuffrida et al., 2017). Similar work has been conducted where the cGAN's labels were masks of manually segmented *Arabidopsis* leaves (Zhu et al., 2018). New segmented masks were then generated and used as labels to the cGAN. The approach for data augmentation by GAN generation we explored differed from these attempts, as we did not wish to have to generate plausible masks to train a cGAN on as in (Zhu et al., 2018). Indeed, this is time-consuming, introduces new parameters to the model and requires expertise that is not always available. The architecture we used was capable of generating images and their annotation simultaneously and had therefore a more generative nature than that of (Isola et al., 2017). Our model was similar to works where GANs were trained on image-annotation pairs (Neff et al., 2017). The experiment for which data was generated in their case was however quite different, as it was mainly showed that for the concerned dataset, replacing part of the training images by simulated images yielded results that were similar to those obtained using only the real images. By contrast here, we showed situations where adding simulated images to the train set could actually improve results.

To sum up, our paper focuses on novel data augmentation strategies applied to boost the supervised segmentation of apple scab observed at a canopy level. No prior work has been done on apple scab segmentation at the canopy level. Thus, we compare segmentation results when training on a "full" training dataset, *i.e.* a training dataset

where adding new annotated images does not significantly improve results to *a reduced part of this training dataset augmented by our strategies*. This course of action is illustrated in Fig. 1. The rest of the article is organized as follows. Our methodology is presented in section 2. Results of the experiments are presented in section. 3 and discussed in section. 4.

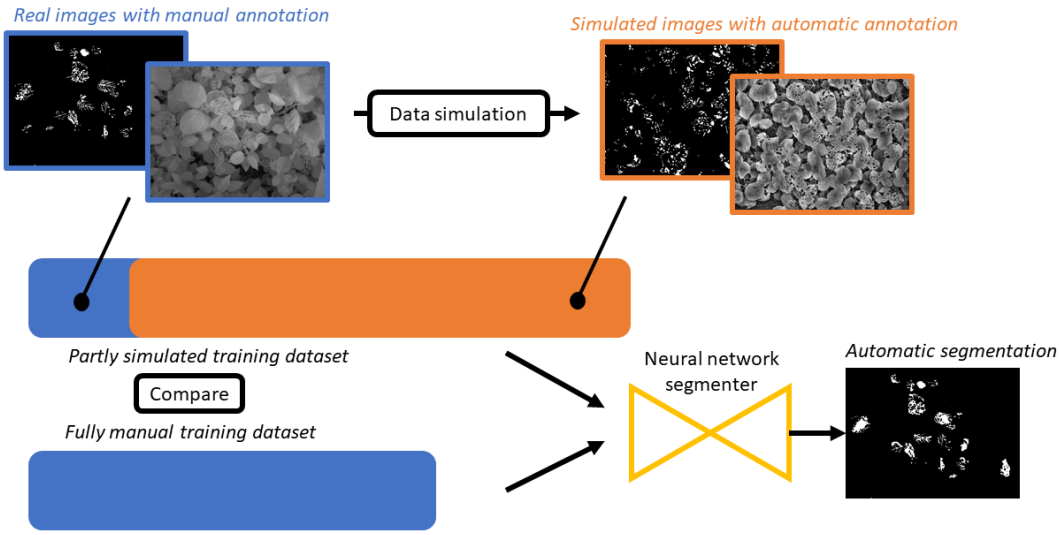


Figure 1: Global illustration of the work. We simulated annotated data from a small set of real annotated images through various simulators. We tested if simulated data increased segmentation score, and also compared this hybrid dataset to a "full" dataset with more real images - whose creation required more manual annotation than for the hybrid dataset.

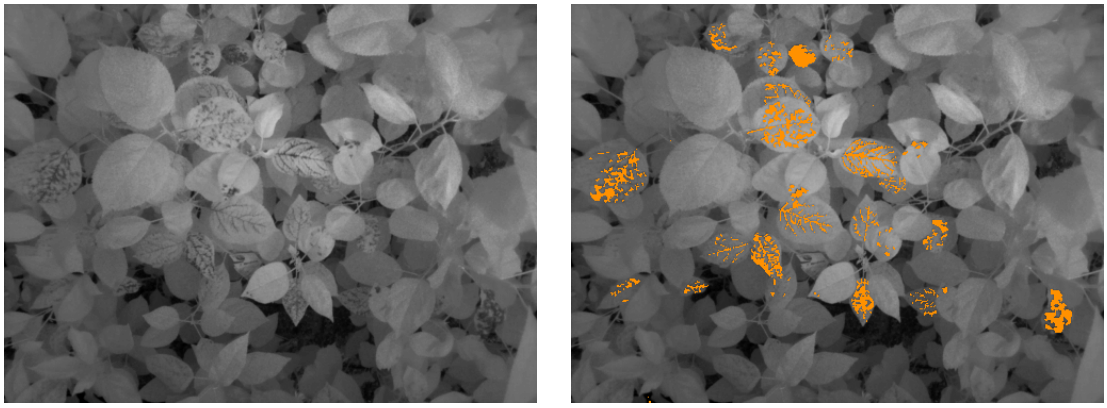
2. Material and methods

2.1. Data acquisition

Images of apple plants were acquired in April 2018 at INRA Angers greenhouses. The camera used was developed by company Carbon Bee, capable of acquiring images both in visible and infrared light. The camera was held by hand at 2m above the ground, facing downwards. The apple plants in the camera's field of view were set on trays and peaked at about 1.5m from the ground. Acquisitions were done around noon and the camera gain was set for an optimal dynamic range without saturation. A dataset of 11 infrared (IR) images of 2592×1944 pixels, representing apple (*Malus* $\times$ *domestica Borkh*) plants inoculated with scab (Fig. 2) was acquired. While the number of images may seem small for a deep learning-based approach, it should be noted that each image represented 50 apple plants, which led to several hundred visible leaves, among which 15 to 40 were afflicted with scab. We annotated this dataset using GIMP by assigning either a "scab" or a "not scab" label to each pixel as visible in Fig. 2. This dataset presented several challenges from a computer vision point of view:

- The segmentation task at hand concerned small local structures to be localized in an image several orders of magnitudes larger than them. Hence, structural information, such as texture, was somewhat limited as scab lesions were captured from an important distance compared to their size.
- Localization of scabbed leaves was fairly easy for human eyes, but a pixelwise localization was difficult because of a "gradient" aspect of scab. The amount of fungi on apple scab spots was indeed more concentrated at the center of these spots and this created a fading of the contrast. Thus, pixelwise annotation was very challenging even for an agronomic expert, and we expected it to to be a challenge for segmentation networks as well.
- Because of the difference in reflectance that scab lesions cause on leaves, scab could seem easily detectable at first glance. However, some factors complicated this distinction: other structures such as leaf veins also caused contrasted areas on the leaves. Moreover, the multi-layer structure of the scene led to leaves that showed very different gray level intensities depending on their distance from the soil and occlusions by other leaves.

- Images were unevenly illuminated due to spatial non-homogeneity in natural lighting. Moreover, they suffered from a slight vignetting effect, meaning more light reached the sensor's center than its borders, leading to images with dark borders.



(a) An infrared image of an apple tree canopy afflicted by apple scab.

(b) Manual annotation (orange) of scab on the image (a).

Figure 2: Illustration of plant disease and observation scale considered in this article for the comparative test of data augmentation strategies.

2.2. Network training

This section presents the segmentation network and training specifics we used for all of our segmentation experiments.

2.2.1. Architecture

We used the SegNet segmentation network (Badrinarayanan et al., 2017). While not being state-of-the-art for segmentation tasks anymore (Cordts et al., 2018), the goal of this paper was to prove the usefulness of simulated images; we only compared results on different train sets and did not strive for absolute best performance. We therefore chose SegNet because robust implementations were available and its inner workings were extensively studied and understood (Noh et al., 2015) compared to more recent networks. For all experiments, we used the following parameters: learning rate $= 10^{-4}$, batch size $= 5$, SGD optimizer with momentum $= 0.99$ and weight decay $= 5 \times 10^{-4}$. These parameters were set by starting with standard values for image segmentation with SegNet, then adjusted through a grid search to prevent overfitting and to yield the best results on the validation set. We implemented class balancing because of the high class imbalance in the dataset (Badrinarayanan et al., 2017).

2.2.2. Dataset preparation

The original images were tiled into sub-images of 64×64 pixels, with no overlap. This was done for computational efficiency and to enable easier generation of images by GAN, which are known to be stable for images of these sizes. During the tiling process, only tiles where at least one pixel had been annotated as "scab" were kept. This strongly helped to reduce class imbalance: in the original dataset, only 2% of pixels were annotated as "scab". After tiling, the scab proportion was 10%. Tiling was not as harmful for segmentation for this dataset as it could have been for other types of images. Images from the dataset represented a global scene of apple plants, while symptoms of interest were fairly local: scab symptoms were independent from plant to plant, and from leaf to leaf. In other words, the general structure of the scene brought almost no information to the local structure of the features of interest. This is not the case in other well-known segmentation datasets such as CamVid (Brostow et al., 2009), where general spatial structure is important. It would therefore have brought no additional information to the network to train on images where several plants were seen together than to train on these plants alone. The dataset was split into train, validation and test sets with respectively 535, 178 and 178 tiles. Out of the 535 tiles of the training set, 107 (20%) were kept as the "reduced dataset" and used in our experiments. The rest of the dataset was kept for comparison of performance with real images. We believed that this reduced quantity of data would be small enough for data augmentation strategies to be useful, and large enough to ensure stable stable segmentation results (in particular regarding overfitting). Moreover, annotating datasets for segmentation is more manageable when these datasets are small. We believe that datasets the size of the "reduced dataset" are close in size to many of the ones used for machine learning in plant sciences.

2.2.3. Training and evaluation

Training was done on the train set and stopped when the score on the validation set had not been improved for 50 epochs. The inference was then carried out on the test set with weights that had yielded the lowest validation loss. The well-known metric "accuracy" (true positives + true negatives over whole population) would have been inadequate to assess the quality of a detection on such an unbalanced dataset: for example, a detection of no scab in an image where 10% of an image were scab led to an accuracy of 90%. Therefore, the metric used for performance evaluation was the F1 score (referred to as simply "score" when discussing the results in section 3), defined as

$$F1 = 2\frac{precision \times recall}{(precision + recall)} \ .$$

This metric is the harmonic mean of precision and recall, defining scab as the class of interest. Precision is the ratio between how many scab pixels the algorithm has marked as such (true positives) and how many pixels the algorithm has marked as scab (true positives + false positives). Recall is the ratio between true positives and how many scab pixels are in the image (true positives + false negatives). A segmentation with a high F1 score therefore has high precision and high recall, meaning few false positives and false negatives. This metric was a relevant choice in such 2-class segmentation cases with heavy class imbalance (Brownlee, 2015).

2.3. Data augmentation strategies

The studied dataset was quite small and segmentation results were quite poor. In this section, we present the different augmentation strategies we implemented for the dataset. Two of them are novel and are described in details in Sections. 2.3.2 and 2.3.3. Our implementation of standard data augmentation, which is a type of data simulation very commonly used in neural network based segmentation, is also presented in section. 2.3.1.

2.3.1. Standard data augmentation

When we implemented standard data augmentation (SDA), we considered the following transformations (Pawara et al., 2017): flip (horizontal or vertical), rotation (90, 180 or 270 degrees), perspective transform by homography, scaling ($\times$0.8 - $\times$1.2) and Gaussian blur (σ = 1 - 2). These transformations were justified by the fact that they could actually occur across images. Rotations and flips were suitable here, as images represented mainly texture and were therefore globally isotropic. Scaling and homography were justified by the fact that images had been acquired with an infrared camera manually held over the apple tree at a distance: the camera could have fluctuated during the acquisition trial. This fluctuation also justified Gaussian blur: the focal of the camera was fixed so that fluctuations of the distance of the camera to plants resulted in a blurring effect. Augmenting a given image meant applying each of these transformations with a certain probability. These probabilities ranged from 0.2 to 0.5. The exact probability depended on the transformation and was meant to reflect the empirical frequency we thought that specific transformation would be useful for the model to generalize better. We devised two ways of augmenting the data:

- offline SDA: We augmented each image in the train set a number n_{aug} of times, multiplying the train set size by n_{aug}.
- online SDA: The train set was left untouched before training starts, but during the training process, transformations were applied to each (original) image at each epoch. It was quite possible that for a relatively small number of epochs, a given image is transformed differently at each epoch. Therefore, the number of different images shown to the network during training with such SDA was actually close to $n_{train} \times n_{epochs}$.

2.3.2. Model-based simulator

We now present the simulator that we devised specifically for generating canopy images. It assumed (i) that the scenes of Fig. 2 could be reproduced by replication of leaves put on top of one another similarly to the "dead leaves model" (Matheron et al., 1975) and (ii) that the apple scab on apple leaves could be characterized as a texture defined by its first and second order statistics. The general pipeline was the following:

1. Randomly draw an isolated apple leaf image from an existing leaf dataset;
2. Generate simulated scab texture and apply it on the leaf;
3. Place the "scabbed" leaf in the scene in a structured way.

The procedure is illustrated in Fig. 3, and the different steps are explained more thoroughly below.

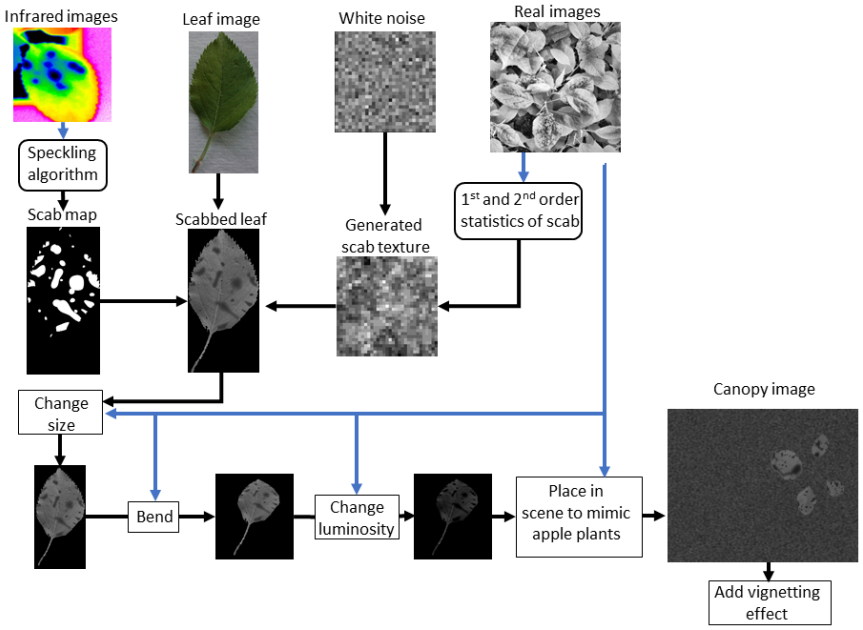


Figure 3: Model-based simulator algorithm. Information from real images and real leaves from a dataset (blue arrows) were used to create a realistic-looking scene. An example output can be seen in Fig. 4.

1. Leaf images: We used the LeafSnap dataset (Kumar et al., 2012) of apple tree leaves set flat and isolated on a uniform background. Images from the train set were used to capture statistics necessary for scab texture synthesis.
   Regarding the texture generation step, we chose to implement a "procedural" process, meaning texture was generated from a filtered Gaussian white noise (McCombs, 2005). The process consisted in fitting the first (mean, standard deviation) and second (autocorrelation) order statistics to the ones recorded on the train set. This simple procedure assumed the apple scab contrast is stationary and captured by a Gaussian random process.
2. Scab pattern: To simulate scab placement on leaves, we studied typical shapes of scab infection in infrared light (Oerke et al., 2011). To reproduce this spatial distribution, we used a speckling algorithm that gave results visually similar to real images. This algorithm is based on taking the Fourier transform of a 2D array of 20×20 random complex numbers and yields a pattern of spots whose sizes are controllable. We then added a Gaussian intensity to these "spots" to produce images closer to real ones.
3. Canopy: Once the leaves were "scabbed", we placed them in the scene, which started off as an artificial image of soil, generated in the same way as scab texture. The goal was to place them with certain occlusion and recovery patterns so that they matched real images as closely as possible. Our approach consisted in simulating apple "plants". We defined a certain number of randomly placed seeds, and iteratively added around each of these seeds a leaf. The placement of a leaf on a given seed was inspired by biological knowledge and observation of real images. In particular, angles between leaves of a plant are well defined because it is known that a plant tries to expose as many as possible to sunlight. This procedure resembled the "dead leaves model", a process proved to simulate images with statistical properties similar to the ones in natural images, such as scale invariance of statistics (Ruderman and Bialek, 1994; Lee et al., 2001).
   To further account for the third dimension of the scene, the top halves of leaves were bent by homography. Moreover, leaves placed later in the iterative process were made bigger and more luminous than former ones, in order to simulate perspective. Luminosity and size changes were handled in an affine way, meaning leaves grew linearly brighter and bigger as they were placed closer to the top. Leaf luminosity was changed by multiplying the intensity of images.

Results can be seen in Fig. 4. Note that the simulator also generated annotation (pixels that represent scab) in addition to the IR images.

Ablation study: To learn more on which parts of the simulation were actually helping on pre-training, we conducted an ablation study, "knocking out" specific features of the simulator described in Section 2.3.2. We then pre-trained the model with these "ablated" images. The list of functions we isolated and knocked out one by one, keeping the others constant, was the following:

- Structure: leaves were kept in layers, but not organized in "apple plants". They were instead randomly positioned and rotated.
- Texture: scab texture was generated using only mean and standard deviation of real scab texture images and not autocorrelation (1), or only autocorrelation without the first order statistics (2).
- Size: leaves from different layers were not resized as a function of the layer they were in.
- Lighting: leaves from different layers' gray level intensity were not changed as a function of the layer they were in.
- Bend: leaves were not bent by homography.
- Gradient: Scab texture was applied to leaves without any "gradient" effect.
- Vignetting: no vignetting effect was added to the image.
- Leaf species : leaves were not apple tree ones (*Malus pumila*), instead mulberry (*Broussonettia papyrifera*) ones.

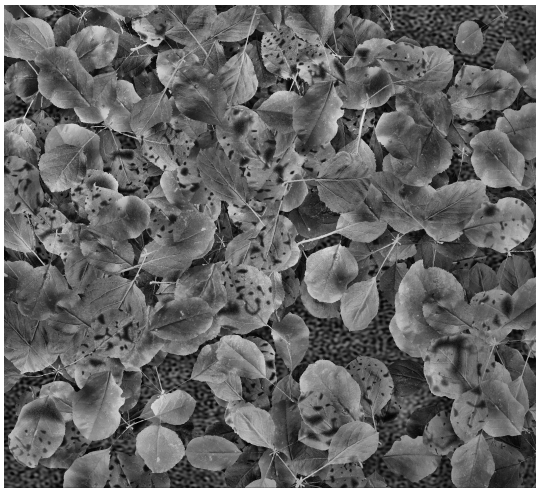


Figure 4: An example of an image coming from the model-based simulator.

2.3.3. GAN simulator

Another way to generate more images from a train set is by using GANs. We used Deep Convolutional GAN (DCGAN) (Radford et al., 2015) architecture, with the Wasserstein GAN (WGAN) (Arjovsky et al., 2017) training process. We tried implementation techniques to improve training stability (Salimans et al., 2016). The only change that improved results was increasing the number of filters in the generator part of the network while leaving the discriminator's number of filters to their default value. Training was done for approximately 100,000 generator iterations, where the loss function did not decrease anymore and simulated images were stable. The difference with traditional image generation by GAN is that the training images we used were concatenated images of IR images (dimension $H \times W \times 1$) and their binary manual annotation (dimension $H \times W \times 1$), yielding an *"annotated image"* (dimension $H \times W \times 2$) for the GAN to train on. Thus, the annotation of a given image was simply encoded as the second channel of that image. There was therefore no need to manually create label images for generation afterward (Zhu et al., 2018): a GAN trained in such a way is capable of generating IR images and their associated annotation. The algorithm is illustrated in Fig. 5.

2.4. Evaluating the simulated images

Once simulated images were created using the simulators described in the previous section, we added them to the train set. We studied two ways of using this additional data to help with the training process. To our knowledge, there has been no research showing that one way of using additional data is more efficient than the other.

- Combining: Create a new train set by adding simulated data to the train set: this seemed like the most natural way of augmenting the train set, as the network would train simultaneously on real and synthesized images. (Ward et al., 2018). The training batches were constrained to contain 50% of real images and 50% of synthesized images. We did not enforce such constraints and rely on the stochasticity of batch formation to generate well-mixed batches.

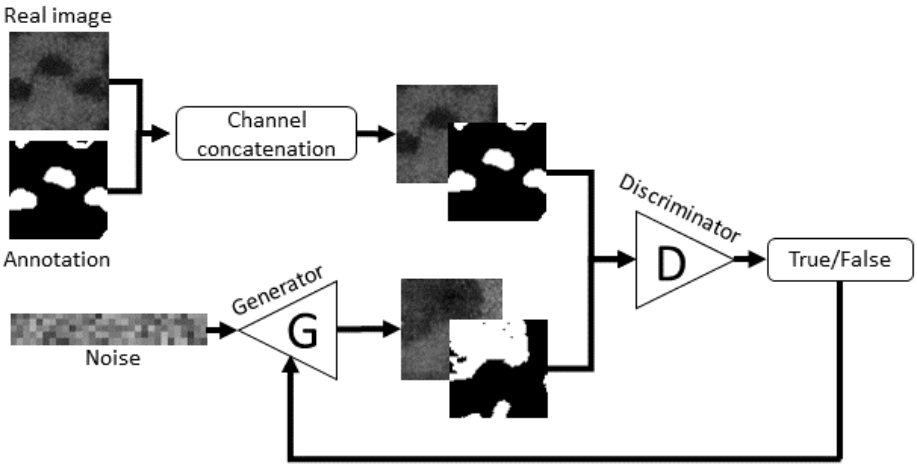


Figure 5: Our GAN-based algorithm description. G and D refer to generator and discriminator network, respectively. Grayscale images are IR images while binary images represent their corresponding annotation.

- Pre-training: Train a network on simulated data and fine-tune it on real images. Using features learned on a dataset to infer on another one, with or without fine-tuning, is one of the earliest practices of neural network training (Yosinski et al., 2014). For example, many applications use weights initialized to ones learned on the ImageNet dataset (Deng et al., 2009; Sharif Razavian et al., 2014). This is based on the idea that most natural images contain similar local patterns, and therefore neurons trained on a generic classification task such as classification on ImageNet are useful for a wide variety of applications. However, we hypothesized that one can achieve better results on a given segmentation task when pre-training on a task close to the final objective instead of on a more generic task.

For all "pre-training" experiments, we used datasets with 5,000 training images. For all "combining" experiences, we tried 4 different simulated dataset sizes corresponding to 0.25, 1, 4 and 8 times the number of real images in the dataset. Results presented in section 3 are shown for the quantities of images that yielded the best scores.

2.5. Comparing with real annotation

To compare the performance of our simulators with real data, we computed segmentation scores of the full dataset of 535 images. More precisely, we created from this full dataset 10 reduced train sets nested into each other, *i.e.* $d_1 \in d_2 \in ... \in d_9 \in d_{full}$, where d_{full} is the dataset of 535 images. We then trained the network on each of the reduced train sets and computed the segmentation scores on the same test set. To complete our analysis, we computed the score on these different train set sizes for different augmentation strategies:

- case 1 : Only pre-training on ImageNet, with no other augmentation (base case);
- case 2 : Pre-training on ImageNet and using (online) SDA, which is still a basic use case of many segmentation routines;
- case 3 : Pre-training on data generated by the model-based simulator and using (online) SDA, which is the scenario we introduced in this paper that provided the highest performance.

3. Results

In this section, we present segmentation scores obtained after training the network presented in Section. 2.2 on the dataset presented in Section. 2, using the various data simulation strategies described in Section. 2.3. Results for a given strategy vary from a run to another, because of the stochasticity of neural network initialization, accentuated by the small dataset sizes. Therefore, all results are averaged over 30 runs, such that score variation between means of batches of 30 runs was under 0.5%. The results are presented in Table 1. With no augmentation whatsoever, the score was 0.424.

Table 1: Segmentation scores for the different dataset augmentation strategies. The ImageNet dataset is typically used for pre-training, and there was no sense in combining it with our dataset as images classification task wildly differ from our case.

Helper dataset	Combining	Pre-training
None	0.424 ± 0.013	
ImageNet	-	0.472±0.009
SDA (offline)	0.559±0.007	0.519±0.006
SDA (online)	0.574±0.010	-
Model-based simulator	0.509±0.005	**0.602**±0.007
GAN-based simulator	0.496±0.011	0.494±0.010

Results for the ablation study of the model-based simulator are presented in Fig. 6.

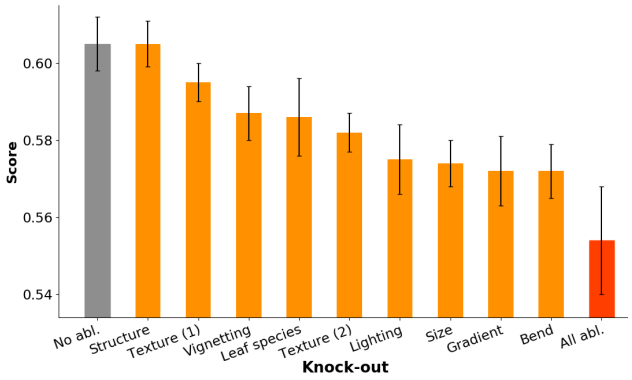


Figure 6: Segmentation scores for the different configurations of the ablation study of the model-based simulator. The X-axis indicates the specific feature of the simulator that was "knocked out" for that experiment. "No abl." refers to an experiment with no ablations (using the full simulator, as in Table 1). "All abl." refers to an experiment where all ablations described in Section 4.2 were applied at once.

We also tested combinations of augmentation strategies: namely adding online data augmentation to train sets partly composed of model-based or GAN-based data.

Table 2: Segmentation scores for the different dataset augmentation strategies combined.

Helper dataset	Combining	Pre-training
Model-based simulator + SDA (online)	0.580±0.004	**0.643**±0.005
GAN-based simulator + SDA (online)	0.599±0.008	0.529±0.007

Results of the comparison of data simulation and actual real data annotation are presented in Fig. 7.

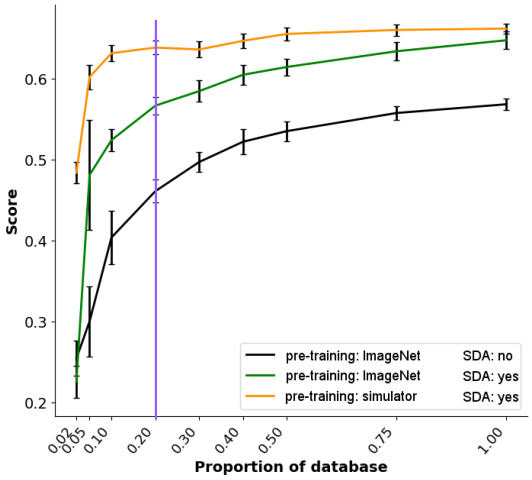


Figure 7: Impact of train set size on segmentation score. The X-axis indicates the proportion of train set kept for training from the full train set (535 images), while validation and test set stay the same size. The Y-axis indicated the segmentation score on the test set. The sub-train set presented in this article up to this point used 0.20 of the train set (marked by the vertical purple line). The different curves correspond to different types of augmentations, as indicated in the legend.

4. Discussion

Let us call the strategy of pre-training on ImageNet the "base case", as it is a widely used technique in machine learning algorithms. The result for the base case was a score of 0.472 (Table 1). This value corresponded to typical values of precision and recall of about 0.533 and 0.424, respectively. For all of our results, we observed relative values of precision and recall close to this case (values are quite close, with precision being higher than recall). To avoid overloading the presentation of the results, precision and recall are not mentioned in Tables 1 and 2. Segmentation suffered from both low precision and low recall. By observing outputs from the model, we noted that low recall came from entire scab lesions being missed. Low precision came from a segmentation "halo" of false positives around detected regions. A possible explanation for this effect was class balancing (see section 2.2), which strongly penalized false negatives to the point where the network became "overcautious" and classified all areas where the decision was difficult (in particular pixels in the aforementioned "gradient zone" (section 2.1)) as "scab", leading to these false positives "halos".. Let us now discuss the impact of the different simulation strategies, compared to this base case. While comparing results, it is worth noting that experiments ran with 30 runs led to a standard deviation of the score typically under 1% (Table 1).

4.1. Standard data augmentation

SDA yielded very convincing progress for this dataset compared to the base case, especially in configurations where it was used directly in combination with our original images. In the offline version, the improvement compared to the base case was 8%. We suspect that with more images, we could have reached similar scores to the online version, which yielded a 10% improvement. This confirmed our intuition that SDA was well adapted to our case: a training image flipped in any direction, rotated in any direction, blurred and/or with changed perspective led to an image that could very much resemble images from the test set. The important positive impact of SDA on plant dataset segmentation is consistent with previous findings (Pawara et al., 2017).

4.2. Model-based simulator

Using the model-based simulator yielded interesting results: the simulated images were efficient in a pre-training scenario (+12%). This showed that images outputted by the simulator were at least somewhat realistic from the networks point of view. However, results when using these simulated images in a combined train set were not much higher than the ones obtained on the base case (+ 3%). The score difference of using these model-based simulated images as pre-training or combining was interesting, as it illustrated the important difference between these two methods of using extra data. In our case, model-based simulated images were very useful in pre-training, meaning they were efficient in leading training of weights "on the right tracks", but were too different from real data to be beneficial if added directly in the train set.

The ablation study (Fig. 6) gives us some insight on the usefulness of our simulator, by showing at which point each of the simulator's features was useful. The "gradient" effect of scab texture and leaf homography seemed to

have the most impact (3% difference with the full simulator) while placing leaves in "plant" structure seems to have been almost useless. It is worth noting that even with all the features tested in this ablation study taken out at once ("All" result in Fig. 6), results were still an 8% gain from the base case. It seems that the mere fact of pre-training on images of real leaves greatly helped the network when training on the scab dataset. We hypothesized that the success of this "simple" pre-training could be explained by the same reasons pre-training on ImageNet helps most training tasks: features learned when pre-training on leaves seem generic enough to be useful to any "plant-related" training tasks.

4.3. GAN-based simulator

Compared to the above augmentations, improvement by the GAN generator was sizably smaller: 2% regardless the way data is used. One possible cause for the smallness of this effect was the difficulty for the GAN to converge to convincing images. Fig. 8 presents different experiments on training the GAN on the train set, which can be seen as intermediary steps to our full training process. While the visual quality of a simulated image did not necessarily strictly correlate with the image's capacity to improve the segmentation process of a real dataset, we believed that the former was a proxy for the latter.

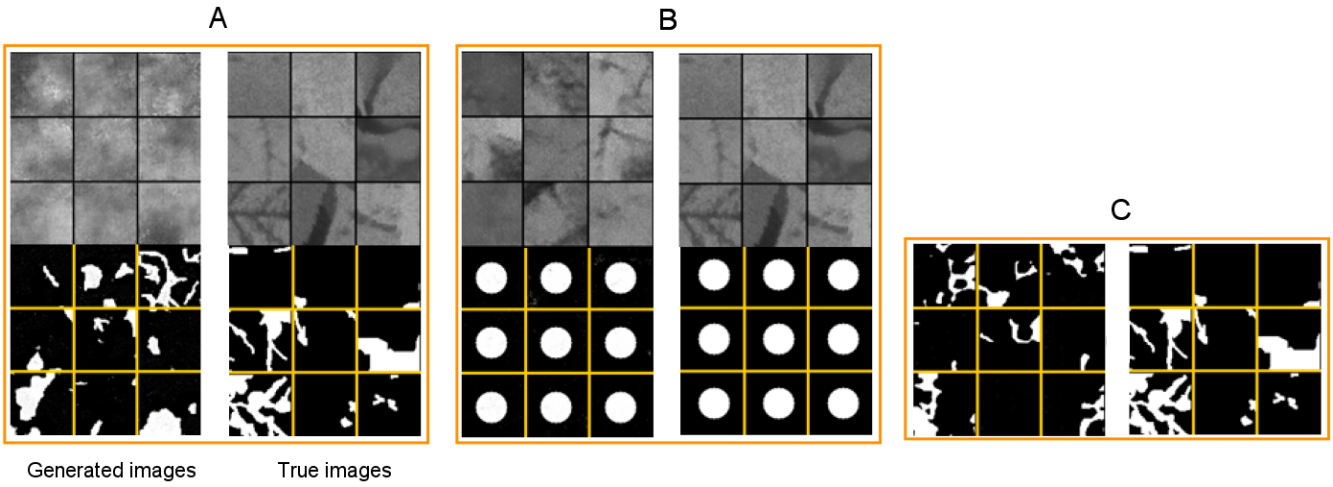


Figure 8: Different experiments (orange squares) to study the difficulties of the GAN based simulator. Each experiment shows 9 examples of original images the GAN trained on (right) and 9 examples of the generator's output after convergence (left). When two blocks of 9 images are presented on top of another (exps. **A** and **B**), they represent an annotated (2-channel) image. Each binary image corresponds to the annotation of the IR image at a similar position. Exp. **A** is a case we present in this article: training on the annotated image. In Exp. **B**, we replaced annotations by "dummy" binary channels, all representing a white circle on black background. Exp. **C** presents GAN training done only on annotations.

The generation presented in this article is shown in Fig. 8 **A**: simulated images seemed relevant because they captured global distribution and structure of IR images and created annotations that seemed realistic when considered on their own. However, they were not satisfactory in the sense that they failed to represent fine structures of original IR images (leaf veins and contours), they had artifacts and annotation structures seemed partly unrelated to their associated IR image. Fig. 8 **B** shows an experiment where the IR images were associated with a "fake" annotation unrelated to the images. Results were visually satisfying, indicating that our GAN algorithm was capable of generating 2 channel images including a grayscale one and a binary one. Fig. 8 **C** presents GAN-simulated images from the binary images only. The GAN seemed to be perfectly capable of generating such binary structures. Thus, the difficulty resided in generating an IR image and a *relevant* binary channel, *i.e.* an annotation, as we saw with the difference between Fig. 8 **A** and **B**. Our intuition was that in order to create these annotated images, the GAN's discriminator's role was much more complex than when creating "simple" grayscale images. When the discriminator had to decide whether an annotated IR image came from a real or fake distribution, it had to judge (among other things) if the two channels were consistent with each other, *i.e.* it considered the image as a whole. In order to be able to decide about that, the discriminator had to find features in the IR channel that could robustly explain pixel values in the annotation channel: in other words, the discriminator had to act as a segmentation network. We believed that this is what made the task difficult for the GAN system and explained poorer results in exp. **A**.

4.4. Multiple augmentations

Our best results were obtained when combining pre-training on the model-based simulator and adding online SDA (Table 2): a 17% increase compared to the base case. This showed that different methods of augmentation could be combined very efficiently.

4.5. Comparing with real annotation

Fig. 7 shows the network's scores on train datasets of varying size, with or without the augmentation techniques discussed in this article. We made the following observations. Firstly, the gains from the different augmentation strategies we observed in this article (vertical purple line on Fig. 7) also appeared for other train set sizes. Next, regardless of the chosen augmentation strategy, segmentation score as a function of train set size was similar to a logarithmic growth. This was quite intuitive : when showing new images to a network learning on a very small train set, these new images were very likely to show new patterns and structures, never seen before by the network. The more images were added, the more the network had seen "everything there is to see", and the less it learned from new images. A corollary of this kind of growth was that scores tended to saturate. With the full train set, we seemed to have not reached full saturation yet. Interestingly, the gap between cases 1 and 2 did not seem to decrease as new images were added. This was a surprising result for us, as we would have expected the curves to converge as we added new images (SDA becoming less and less useful as we cover more and more cases), as it

was the case for the gap between cases 2 and 3. It seemed here that SDA may enable to get on the whole better results compared to adding some new "real" data. Especially for small train set sizes, we saw the usefulness of data augmentation strategies for saving annotation effort. For example, to get a score similar to the one obtained by annotating 150 images (30%) of the train set in case 1, one would need only about 30 images in case 2 and 10 in case 3. Finally, regardless of the type of augmentation, we found these curves to be interesting from a theoretical and practical point of view. Consider the case where one computes one of these curves for a dataset A, and also wishes to work on a dataset B that seems quite similar to A according to certain metrics (in our example, if A is our scab dataset, a similar B dataset could be images of another type of biotic stress on another plant species' leaves for example, with camera position and plant density close to A). Then one could assume that the "data utility curve" of A and B are similar, and therefore infer beforehand the sufficient quantity of B data to annotate for "best possible" segmentation performance. In the case of segmentation annotation, especially in a domain like agronomy where expert intervention is needed, knowing this kind of information can be very valuable.

4.6. Extension to other plant diseases

Many plant diseases show general structure close to the one of apple scab. For example, downy mildew of grapevines, late blight of potatoes and early symptoms of fusarium wilt will resemble small textured lesions appearing on leaves. The GAN-based simulator is in theory an image-agnostic tool, applicable to other segmentation tasks. Since GANs rely on finding some structure in data in a latent space, we believe the results presented here would be reproducible for these other diseases where image structure is close to our scab database. We believe that our model-based simulator could also be extended to these other types of plant diseases characterized by blobs on the leaves. Indeed, contrast, size, spatial distribution and texture of the lesions, size, density, and the species of leaves are all parameters of this simulator which could be tuned.

An extension of our simulator would be to be able to handle color images. While IR imagery was proved to be an appropriate modality for apple scab, standard RGB images can also provide valuable information on a wide range of plant diseases (Mahlein, 2016). In such RGB images, leaf edges are more clearly defined, perspective and leaf veins appear much more distinctly. An interesting perspective would be to provide both RGB and IR images to the network. Such multi-modal images are moreover becoming increasingly easier to acquire thanks to the development of convenient multi-modal image sensors for the plant science domain (Lowe et al., 2017).

5. Conclusion and future works

We have tackled the difficult and open problem in plant science of apple scab segmentation at canopy level. Like many segmentation tasks, the train set was quite small, and with this in mind we designed novel ways of generating new annotated data from the existing train set. This new data helped us achieve a much better segmentation, especially when different types of augmentation were combined.

Acknowledgements

Authors thank the PHENOTIC platform of INRA Angers, member of the PHENOME-EMPHASIS plant phenotyping network together with Charles-Eric Durel and Caroline Denance from INRA Angers for assistance in image acquisition of apple trees with scab in controlled environment. Clément Douarre gratefully acknowledges ANRT for CIFRE PhD funding under 2017/0639.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks. In: International Conference on Machine Learning. pp. 214–223.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39 (12), 2481–2495.

Barth, R., IJsselmuiden, J., Hemming, J., Van Henten, E. J., 2018. Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. Computers and Electronics in Agriculture 144, 284–296.

Belin, É., Rousseau, D., Boureau, T., Caffier, V., 2013. Thermography versus chlorophyll fluorescence imaging for detection and quantification of apple scab. Computers and Electronics in Agriculture 90, 159–163.

Benoit, L., Benoit, R., Belin, É., Vadaine, R., Demilly, D., Chapeau-Blondeau, F., Rousseau, D., 2016. On the value of the kullback–leibler divergence for cost-effective spectral imaging of plants by optimal selection of wavebands. Machine Vision and Applications 27 (5), 625–635.

Benoit, L., Rousseau, D., Belin, É., Demilly, D., Chapeau-Blondeau, F., 2014. Simulation of image acquisition in machine vision dedicated to seedling elongation to validate image processing root segmentation algorithms. Computers and electronics in agriculture 104, 84–92.

Bowen, J. K., Mesarich, C. H., Bus, V. G., Beresford, R. M., Plummer, K. M., Templeton, M. D., 2011. Venturia inaequalis: the causal agent of apple scab. Molecular Plant Pathology 12 (2), 105–122.

Brostow, G. J., Fauqueur, J., Cipolla, R., 2009. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters 30 (2), 88–97.

Brownlee, J., 2015. Tactics to combat class imbalance in your machine learning dataset. `https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`, accessed: 2018-10-14.

Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Belin, É., Chapeau-Blondeau, F., 2012. On the use of depth camera for 3d phenotyping of entire plants. Computers and Electronics in Agriculture 82, 122–127.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2018. Semantic understanding of urban street scenes. `https://www.cityscapes-dataset.com/benchmarks/`, accessed: 2018-10-12.

Delalieux, S., Somers, B., Verstraeten, W., Van Aardt, J., Keulemans, W., Coppin, P., 2009. Hyperspectral indices to diagnose leaf biotic stress of apple plants, considering leaf phenology. International Journal of Remote Sensing 30 (8), 1887–1912.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 248–255.

Di Cicco, M., Potena, C., Grisetti, G., Pretto, A., 2017. Automatic model based dataset generation for fast and accurate crop and weeds detection. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5188–5195.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H., 2018. Synthetic data augmentation using gan for improved liver lesion classification. In: Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on. IEEE, pp. 289–293.

Giuffrida, M. V., Scharr, H., Tsaftaris, S. A., 2017. Arigan: Synthetic arabidopsis plants using generative adversarial network. In: Proceedings of the 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), Venice, Italy. pp. 22–29.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. Vol. 1. MIT press Cambridge.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., 2017. Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 5967–5976.

Kamilaris, A., Prenafeta-Boldú, F. X., 2018. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture 147, 70–90.

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I., Soares, J. V. B., October 2012. Leafsnap: A computer vision system for automatic plant species identification. In: The 12th European Conference on Computer Vision (ECCV).

Lee, A. B., Mumford, D., Huang, J., 2001. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. International Journal of Computer Vision 41 (1-2), 35–59.

Lowe, A., Harrison, N., French, A. P., 2017. Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. Plant methods 13 (1), 80.

Mahlein, A.-K., 2016. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. Plant Disease 100 (2), 241–251.

Matheron, G., Matheron, G., Matheron, G., Matheron, G., 1975. Random sets and integral geometry.

McCombs, S., 2005. Intro to procedural textures. `http://www.upvector.com/?section=Tutorials&subsection=IntrotoProceduralTextures`, accessed: 2018-10-10.

Minervini, M., Scharr, H., Tsaftaris, S. A., 2015. Image analysis: the new bottleneck in plant phenotyping [applications corner]. IEEE signal processing magazine 32 (4), 126–131.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Neff, T., Payer, C., Štern, D., Urschler, M., 2017. Generative adversarial network based synthesis for supervised medical image segmentation.

Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1520–1528.

Oerke, E.-C., Fröhling, P., Steiner, U., 2011. Thermographic assessment of scab disease on apple leaves. Precision Agriculture 12 (5), 699–715.

Pawara, P., Okafor, E., Schomaker, L., Wiering, M., 2017. Data augmentation for plant classification. In: International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, pp. 615–626.

Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., Bulat, A., Tzimiropoulos, G., Wells, D. M., Murchie, E. H., et al., 2017. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. GigaScience.

Pradal, C., Dufour-Kowalski, S., Boudon, F., Fournier, C., Godin, C., 2008. Openalea: a visual programming and component-based software platform for plant modelling. Functional plant biology 35 (10), 751–760.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Ruderman, D. L., Bialek, W., 1994. Statistics of natural images: Scaling in the woods. In: Advances in neural information processing systems. pp. 551–558.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242.

Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 806–813.

Ubbens, J., Cieslak, M., Prusinkiewicz, P., Stavness, I., 2018. The use of plant models in deep learning: an application to leaf counting in rosette plants. Plant methods 14 (1), 6.

Ward, D., Moghadam, P., Hudson, N., 2018. Deep leaf segmentation using synthetic data. arXiv preprint arXiv:1807.10931.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328.

Zhu, Y., Aoun, M., Krijn, M., Vanschoren, J., Campus, H. T., 2018. Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. Computer Vision Problems in Plant Phenotyping (CVPPP2018).