



## OPEN ACCESS

## EDITED BY

Ribana Roscher,  
Institute for Bio- and Geosciences Plant  
Sciences (IBG-2), Germany

## REVIEWED BY

Manya Afonso,  
Wageningen University and  
Research, Netherlands  
Kamil Dimililer,  
Near East University, Cyprus

## \*CORRESPONDENCE

Christopher J. Henry  
✉ ch.henry@uwinnipeg.ca

RECEIVED 05 April 2023

ACCEPTED 05 June 2023

PUBLISHED 06 July 2023

## CITATION

Krosney AE, Sotoodeh P, Henry CJ, Beck MA  
and Bidinosti CP (2023) Inside out: transforming  
images of lab-grown plants for machine  
learning applications in agriculture.  
*Front. Artif. Intell.* 6:1200977.  
doi: 10.3389/frai.2023.1200977

## COPYRIGHT

© 2023 Krosney, Sotoodeh, Henry, Beck and  
Bidinosti. This is an open-access article  
distributed under the terms of the [Creative  
Commons Attribution License \(CC BY\)](#). The use,  
distribution or reproduction in other forums is  
permitted, provided the original author(s) and  
the copyright owner(s) are credited and that  
the original publication in this journal is cited, in  
accordance with accepted academic practice.  
No use, distribution or reproduction is  
permitted which does not comply with these  
terms.

# Inside out: transforming images of lab-grown plants for machine learning applications in agriculture

Alexander E. Krosney<sup>1,2</sup>, Parsa Sotoodeh<sup>3</sup>, Christopher J. Henry<sup>3\*</sup>,  
Michael A. Beck<sup>3</sup> and Christopher P. Bidinosti<sup>2,3</sup>

<sup>1</sup>Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada, <sup>2</sup>Department of Physics, University of Winnipeg, Winnipeg, MB, Canada, <sup>3</sup>Department of Applied Computer Science, University of Winnipeg, Winnipeg, MB, Canada

**Introduction:** Machine learning tasks often require a significant amount of training data for the resultant network to perform suitably for a given problem in any domain. In agriculture, dataset sizes are further limited by phenotypical differences between two plants of the same genotype, often as a result of different growing conditions. Synthetically-augmented datasets have shown promise in improving existing models when real data is not available.

**Methods:** In this paper, we employ a contrastive unpaired translation (CUT) generative adversarial network (GAN) and simple image processing techniques to translate indoor plant images to appear as field images. While we train our network to translate an image containing only a single plant, we show that our method is easily extendable to produce multiple-plant field images.

**Results:** Furthermore, we use our synthetic multi-plant images to train several YoloV5 nano object detection models to perform the task of plant detection and measure the accuracy of the model on real field data images.

**Discussion:** The inclusion of training data generated by the CUT-GAN leads to better plant detection performance compared to a network trained solely on real data.

## KEYWORDS

digital agriculture, agriculture 4.0, deep learning, convolutional neural networks, generative adversarial networks, data augmentation, image augmentation

## 1. Introduction

Machine learning (ML) tasks are often limited by the availability and quality of training data for a given model (LeCun et al., 2015; Goodfellow et al., 2016). To enable ML-based applications in agriculture—such as the automatic detection and classification of plants or crop health monitoring, say—ultimately requires large quantities of labeled image data with which to train deep neural networks (DNNs) (Lobet, 2017; Liakos et al., 2018; Wäldchen et al., 2018; Lu et al., 2022). It is the present lack of such data, and the challenge of generating it, that may ultimately limit the broad application of such techniques across the immense variety of crop plants. Lobet (2017), for example, emphasize that this process “is hampered by the difficulty of finding good-quality ground-truth datasets.” Similar sentiments are echoed in general reviews (Liakos et al., 2018; Wäldchen et al., 2018; Lu et al., 2022), as well as in publications on specific applications such as weed detection (Binch and Fox, 2017; Bah et al., 2018; Bosilj et al., 2018) and high-throughput phenotyping (Fahlgren et al., 2015; Singh et al., 2016; Gehan and Kellogg, 2017; Shakoor et al., 2017; Tardieu et al., 2017; Giuffrida et al., 2018). The difficulty of generating or collecting plant-based image data for agricultural applications is further exacerbated by the many differences in growing

conditions and physical dissimilarities between any two plants, even for those belonging to the same genotype. Covering a wide variety of phenotypes with a sufficient volume of labeled training data is a task of massive scope. This challenge is further impeded by the requirement of expert knowledge that is often necessary to accurately label plant data (for example, when it comes to the distinction between oats and wild oats) (Beck et al., 2021).

Image transformation and synthesis through the use of generative adversarial networks (GANs) is gaining interest in agriculture as a means to expedite the development of large-scale, balanced and ground-truthed datasets (Lu et al., 2022). GANs were originally used for creating synthetic MNIST digits, human faces, and other image types (Goodfellow et al., 2014) and have proven successful for a wide variety of quite remarkable image translation problems, including transforming a horse to/from a zebra, a dog to/from a cat, and a summer scene to/from a winter scene (Isola et al., 2016; Zhu et al., 2017; Park et al., 2020). In the agricultural domain, GANs have been applied to areas such as plant localization, health, weed control, and phenotyping (Lu et al., 2022). Some specific examples that demonstrate the promise of GANs in agriculture include the improvement of plant image segmentation masks (Barth et al., 2018), disease detection in leaves (Zeng et al., 2020; Cap et al., 2022), leaf counting (Giuffrida et al., 2017; Zhu et al., 2018; Kuznichov et al., 2019), modeling of seedlings (Madsen et al., 2019), and rating plant vigor (Zhu et al., 2020).

The contribution of this work is not an automated task or application to solve a particular agricultural problem, but rather the development of a method to translate real indoor images of plants to appear in field settings. As such, this work serves as an important precursor to the development of such applications. This approach can in principle be used to create bespoke, labeled data sets to support the training needs of a wide variety of ML tasks in agriculture. For example, appropriately constructed collages of two or more species on a soil background could ultimately be used to synthesize very large numbers of images of crop plants interspersed with weeds, resulting in labeled, ground-truthed datasets suitable for developing ML models for automated weed detection. For this initial study, however, testing is limited to simpler demonstrations of object detection.

The impetus for this work comes from our previous development of an embedded system for the automated generation of labeled plant images taken indoors (Beck et al., 2020). Here, a camera mounted a computer controlled gantry system is used to take photographs of plants against blue keying fabric from multiple positions and angles. Because the camera and plant positions are always known, single-plant images can be automatically cropped and labeled. In addition to this, we have collected outdoor images of plants and soil (Beck et al., 2021), which at present must be cropped and annotated by hand. Belonging to both datasets are four crop species: canola, oat, soybean, and wheat. The presence of these plants in both datasets provides the opportunity for outdoor image synthesis through image-to-image translation via GANs (Isola et al., 2016). Our goal, then, is to create fully labeled training datasets that are visually consistent with real field data. This procedure eliminates the need for manual labeling of outdoor grown plants (which is time-consuming and prone to error) while being scalable and adaptable to new environments (e.g., different

soil backgrounds, plant varieties, or weather conditions). The creation of one's own datasets could improve the accuracy of plant detection and other models in a real field setting.

This paper is structured as follows. Section 2 provides an overview of the GAN architecture used in our image translation experiments and describes the construction process for the GAN training datasets. Section 3 presents visual results for several single-plant translation experiments and discusses the benefits and limitations of each training dataset. Section 4 describes our method for producing augmented outdoor multi-plant images with automated labeling. Section 5 presents plant detection results achieved using a YoloV5 nano model trained on our augmented datasets. Section 6 concludes the paper and discusses potential extensions to the image synthesis methods.

## 2. Materials and methods

Many GAN architectures require the availability of paired data for training. In our case, an image pair would consist of a plant placed in front of a blue screen and an identical plant, in the same location of the image, placed in soil. Such image pairs are difficult to obtain in large volumes and instead we focus on GAN architectures that can train on unpaired data (Zhu et al., 2017), such as the examples shown in Figure 1, which are taken from our indoor and outdoor plant datasets.<sup>1</sup>

Some GANs, such as CycleGAN, consist of two-sided networks that not only translate an image from one domain to another but perform the reverse translation as well (Zhu et al., 2017). For our specific problem, we are interested in single-directional translation and only consider one-sided networks to reduce training duration and model sizes.

As a result, the approach taken here follows that of contrastive unpaired translation (CUT) as presented in Park et al. (2020). For this, one considers two image domains  $X$  and  $Y$  (with samples  $\{x_i\}_{i=1}^{N_x} \subseteq X$  and  $\{y_j\}_{j=1}^{N_y} \subseteq Y$ ) and seeks to find a function that takes a sample from the domain  $X$  and outputs an image that can plausibly come from the distribution  $Y$ .

Generative adversarial networks typically consist of two separate networks that are trained simultaneously. The generator  $G$  learns the mapping  $G: X \rightarrow Y$  and the discriminator  $D$  is trained to differentiate between the real images  $y$  of domain  $Y$  and the fake images  $G(x) = \hat{y}$  produced by the generator. Note that we refer to the real and fake images of domain  $Y$  as  $y$  and  $\hat{y}$ , respectively. The discriminator returns a probability in  $[0.0, 1.0]$  that the input image came from the distribution  $Y$ . Effectively, the generator is trained to produce images that fool the discriminator by minimizing the adversarial loss (Goodfellow et al., 2014; Isola et al., 2016; Giuffrida et al., 2017; Zhu et al., 2017, 2018; Park et al., 2020)

$$\mathcal{L}_{GAN}(G, D, X, Y) := \mathbb{E}_{y \sim Y} \log D(y) + \mathbb{E}_{x \sim X} \log (1 - D(G(x))). \quad (1)$$

<sup>1</sup> At the time of writing, the indoor dataset contains over 1.2 million labeled images of 14 different crops and weeds commonly found in Manitoba, Canada, while the outdoor dataset contains 540,000 still images extracted from video footage of five different common crops of this region. The datasets can be made available upon request.

A visual overview of a GAN structure is given in Figure 2.

The generator is composed of two networks that are applied sequentially to an image. The first half, the encoder  $G_{enc}$ , receives the input and constructs a feature stack, primarily through down-sampling operations. The second half, the decoder  $G_{dec}$ , takes a feature stack and constructs a new image, through up-sampling operations. Here, we have that  $G(x) = G_{dec}(G_{enc}(x)) = \hat{y}$  (Park et al., 2020).

Contrastive unpaired translation is a GAN architecture that enables one-sided image-to-image translation (Park et al., 2020). In addition to the adversarial loss, which is dependent on the networks  $G$  and  $D$ , CUT provides the PatchNCE loss and feature network  $H$ . The PatchNCE loss is used both to retain mutual information between the input image  $x$  and output  $\hat{y}$  as well as to enforce the identity translation  $G(y) = y$  (Park et al., 2020). The feature

network is defined as the first half of the generator, the encoder, plus a multi-layer patchwise (MLP) network with two layers and is used to encode the input and output images into feature tensors. Patches from the output image  $\hat{y}$  are sampled, passed to the feature network, and compared to the corresponding (positive) patch from the input as well as  $N$  other (negative) patches from the input image. The process is shown in Figure 3 (Park et al., 2020).

Since  $G_{enc}$  is used to translate a given image, its feature stack is readily available, with each layer and spatial position corresponding to a patch in the input image. We select  $L$  layers from the feature map and pass each layer through the patchwise network to produce features  $\{z_l\} = \{H_l(G_{enc}^l(x))\}_L$  where  $G_{enc}^l$  is the output of the  $l$ -th layer. The feature  $z_l^s$  then represents the  $s$ -th spatial location of the  $l$ -th layer and  $z_l^{S \setminus s}$  represents all other locations. The PatchNCE loss is given by Park et al. (2020) as

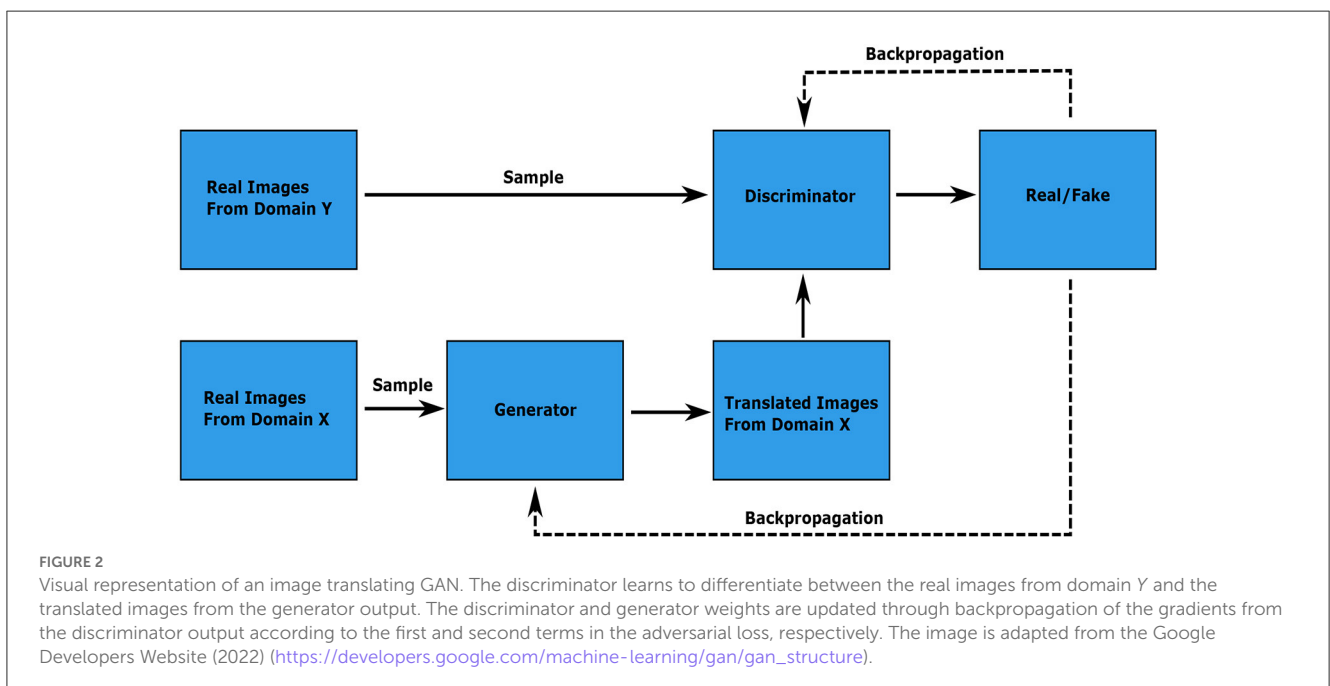
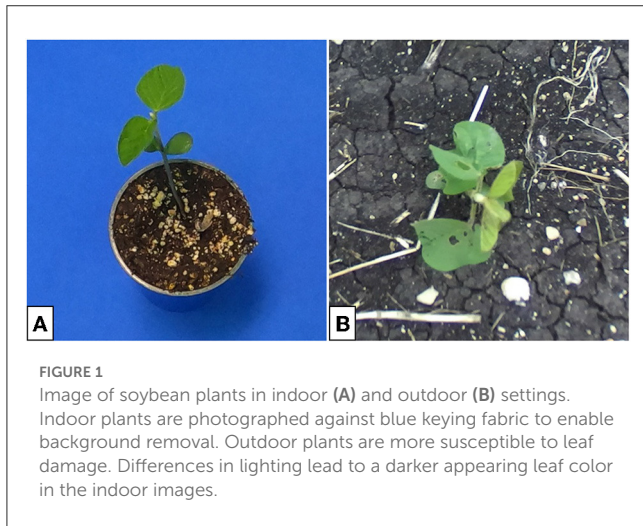
$$\mathcal{L}_{PatchNCE}(G, H, X) := \mathbb{E}_{x \sim X} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{z}_l^s, z_l^s, z_l^{S \setminus s}), \quad (2)$$

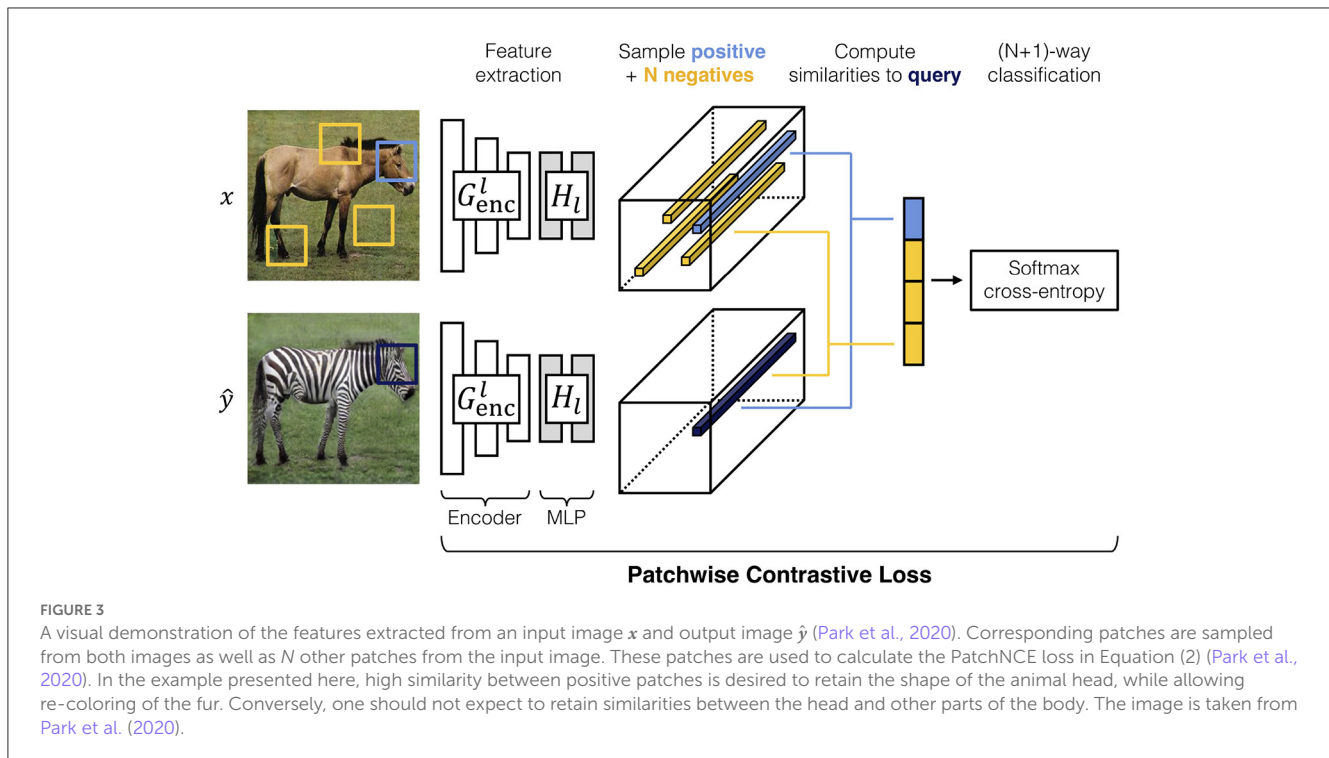
where the sums are taken over all desired layers  $l$  and spatial locations  $s$  within each layer,

$$\ell(z, z^+, z^-) := -\log \left[ \frac{\exp(z \cdot z^+ / \tau)}{\exp(z \cdot z^+ / \tau) + \sum_{n=1}^N \exp(z \cdot z_n^- / \tau)} \right], \quad (3)$$

where  $N$  is the number of negative samples and the temperature  $\tau = 0.07$  scales the magnitude of penalties on the negative samples (Park et al., 2020). The overall loss used for network training is

$$\mathcal{L}(G, D, H, X, Y) = \mathcal{L}_{GAN}(G, D, X, Y) + \lambda_X \mathcal{L}_{PatchNCE}(G, H, X) + \lambda_Y \mathcal{L}_{PatchNCE}(G, H, Y), \quad (4)$$





where the weighing factors  $\lambda_X = \lambda_Y = 1$  when using the default CUT options (Park et al., 2020). The first PatchNCE loss term is used to retain mutual information between the input image  $x$  and output  $\hat{y}$ . The second is used as an identity loss term, where translating the input image  $y$  should produce  $y$  as the output.

The CUT training scripts provide several network definitions for the generator, discriminator, and feature network. For the work in this paper, we use the default options where the generator is a ResNet-based architecture that consists of 9 ResNet blocks between up and down-sampling layers and the discriminator is a  $70 \times 70$  PatchGAN network that originates from the work of Isola et al. (2016).

The remainder of this section provides an overview of the datasets used to train several CUT generators.

## 2.1. Target domain—Outdoor image dataset

Generator training requires a dataset of outdoor images to represent the target domain  $Y$  for image translation. Our outdoor dataset was constructed by sampling from the 540,000 available images in the outdoor image database (Beck et al., 2021). The images are individual frames from videos taken with a camera mounted to a tractor while traveling through a field. The full-resolution images ( $2208 \times 1242$  px) often contain several plants with unknown locations and must be cropped by hand to obtain single-plant photos suitable for image translation. Manual cropping is a time-consuming process and, in general, limits our overall training dataset size. For initial experiments, 64 hand-cropped single-plant field images were used to construct the target dataset. Larger-scale experiments that include image translation of several species were also performed with 512 single-plant field images. Table 1

summarizes the relevant parameters for all training datasets. Example multi- and cropped single-plant outdoor images are shown in Figure 4.

The outdoor plant datasets used for model training contain several single-plant images of canola, oat, soybean, and wheat. The cropping bounds are varied to provide data with plants of several sizes relative to the image, this helps to prevent the generator from increasing or decreasing the size of the plant in the image during translation. Similarly, plant locations (e.g., center, top-left, bottom-right, etc.) are varied across the single-plant images to minimize positional drift. Additional images sampled from this dataset can be seen in Figures 8A–10A.

## 2.2. Input domain—Cropped lab image dataset

The first dataset to be used as the input domain  $X$  to the generator consists of several indoor single-plant images with a blue screen background. These images are provided by the EAGL-I system, as described in Beck et al. (2020). EAGL-I employs a GoPro Hero 7 camera mounted on a movable gantry capable of image capture from positions that vary in all three spatial dimensions. Additionally, the gantry includes a pan-tilt system to provide different imaging angles. For the purpose of this work, we attempt only to translate top-down images, thus we include images only where the camera is perpendicular to the floor, within a range of  $\pm 10^\circ$ .

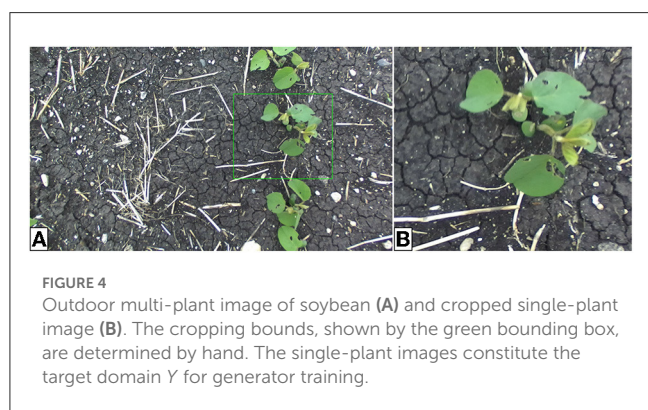
Similar to the previous section, the full-resolution images ( $4000 \times 3000$  px) contain several plants and must be cropped to obtain single-plant photos for training. However, in this case, both the



TABLE 1 Construction parameters for each training dataset.

Dataset name	$N_x$	$N_y$	Species	Age (days)	$N_{\text{backgrounds}}$	$S_{\text{min}}$	$S_{\text{max}}$	Figure
Cropped lab	64	64	Soybean	10-40	N/A	1.00	1.00	Figure 8
Composites	64	64	Soybean	10-40	32	0.50	0.85	Figure 9
Color-corrected composites 1	64	64	Soybean	10-40	32	0.50	0.85	Figure 10
Color-corrected composites 2	512	512	All	Varies	128	0.50	0.85	Figure 11

For the *Cropped Lab* dataset, background replacement is not applicable since composite images are not created. Furthermore, the plant always makes up the entire image, so the scale is always 1.00. The *Color-Corrected Composites 2* dataset contains each of the four available species; canola, oat, soybean, and wheat. These plants have minimum-maximum age ranges of 10–40, 0–365, 10–40, and 0–365 days, respectively.



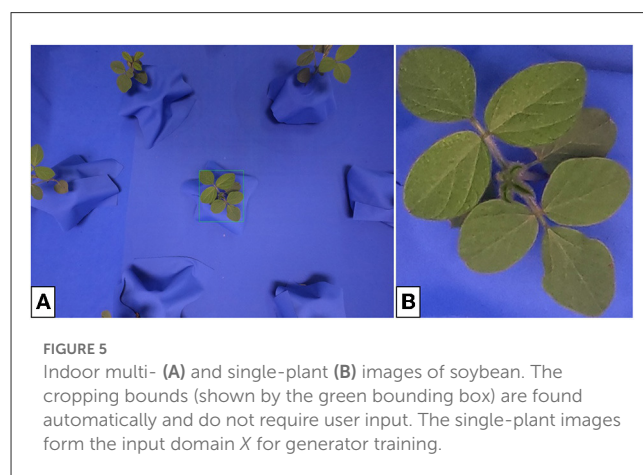
plant and camera positions within the setting are known, and loose bounding boxes are found through geometric calculation. Tighter bounding boxes are obtained through an algorithm given in Appendix 1 (Supplementary material), avoiding the need for hand-cropping. Figure 5 shows an example of both a multi- and single-plant lab image.

To construct our dataset for generator input, we use 64 single-plant lab images as shown in Figure 5B. It is expected that a translated image would not only contain a plant with characteristics that reflect the outdoor plant domain, but also contain soil that has been generated in place of the blue screen background.

### 2.3. Input domain—Composite image dataset

As an alternative to the single-plant indoor photos of Section 2.2, image processing techniques can be used to remove the blue screen in a single-plant image and replace it with a real image of soil. These images, henceforth known as composite images, ideally require minimal translation to the image background and instead allow the generator to primarily translate plant characteristics. Such a network would be beneficial as it provides the user with the ability to influence the appearance of the background, even after passing through the generator.

Soil backgrounds are randomly sampled from a set of images to provide sufficient variation in the training data. These backgrounds are hand-cropped from real outdoor data, so the soil is visually consistent with our target domain. Typically, 32 soil backgrounds



were used for datasets composed of 64 composites. See Table 1 for the exact number of backgrounds used to construct each dataset.

Given an indoor multi-plant image and soil background, the steps for composite formation are listed below. This process is fully-automated through scripts written in Python that utilize the OpenCV and NumPy libraries for image processing. Example images that depict each step are shown in Figure 6.

1. Mask and create bounding boxes using a multi-plant image from the indoor plant database. Blue screen usage makes background removal a trivial problem that can be solved by image thresholding (Beck et al., 2020).
2. Crop the multi-plant image and masked multi-plant image to create single-plant images.
3. Remove the blue screen in the single-plant image with the single-plant mask.
4. Randomly re-scale the single-plant image size relative to the final output size. We choose the minimum and maximum scale values to be  $S_{\text{min}} = 0.50$  and  $S_{\text{max}} = 0.85$ , respectively. The image size required for the CUT generator is  $256 \times 256$  px, so the plant is scaled relative to this size. Plant scale is calculated by dividing the longest side of the single-plant image by the output image size. For example, if the single-plant image has dimensions  $150 \times 200$  px and the output image has dimensions  $256 \times 256$  px, then the scale is  $200/256 = 0.78$ . Finally, the resized image is padded with black pixels so the final image has size  $256 \times 256$

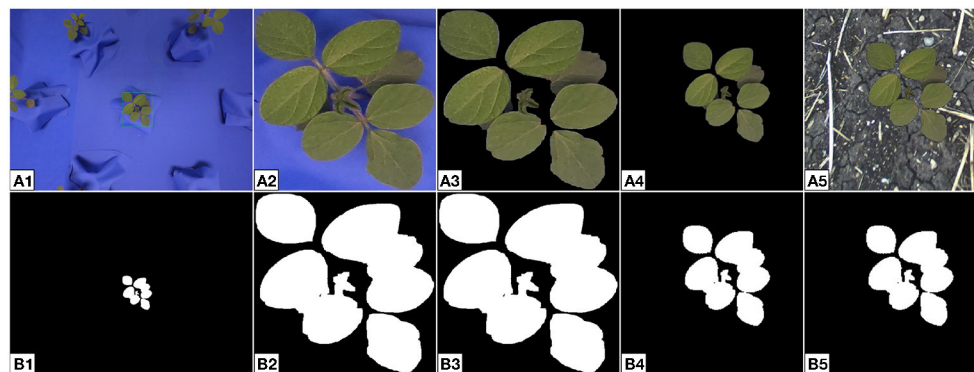


FIGURE 6

Intermediate images created during composite image formation (A) and their associated binary masks (B). The multi-plant image (A1, B1) is cropped to obtain a single-plant image (A2, B2). The single-plant binary mask is used to remove the blue screen in the single-plant image (A3, B3). The single-plant image is then resized and padded to give the effect of random placement within the final image (A4, B4). A soil background is joined to the resized image to create the final composite (A5, B5). Images such as (A5) are used to form the input domain  $X$  for generator training.

px. Random padding gives the effect of random placement of the plant within the composite.

- Combine the resized plant image with a randomly selected soil background.

In many cases, differences in lighting between indoor and outdoor images leave little contrast between the plant and soil background in a composite image. In preliminary generator training experiments, insufficient contrast could lead to the generator translating plant leaves to appear as soil and constructing a fully-synthetic plant in a different region of the image. Such an occurrence is detrimental for creating synthetic object detection data as one loses the ability to accurately provide a bounding box for the plant. In the datasets described below, color-correction is used during composite formation to match the plant color with real field plants and increase the contrast between the plant and soil to allow the generator to maintain semantic information after domain translation. Color-correction techniques such as histogram matching were explored but were not used in favor of simple mean matching since one would still expect the generator to adjust plant color. The chosen color-correction process is described as follows.

Given an  $(M \times N \times 3)$  CIELAB single-plant image  $\mathbf{x}$  and the associated  $(M \times N)$  binary mask  $\mathbf{m}$ , we want to correct the pixels belonging to the plant to have color that is consistent with the real outdoor plants in the domain  $Y$  and leave all other pixels unchanged. To do this, we first determine the average value for the  $k$ -th channel of  $\mathbf{x}$ , only where the mask is true, given by

$$\bar{x}_k = \frac{\sum_{i=1}^M \sum_{j=1}^N x_{ijk} m_{ij}}{\sum_{i=1}^M \sum_{j=1}^N m_{ij}}. \quad (5)$$

Note that Equation (5) is simply the weighted average for all pixels of a given channel. The weight of each pixel is given by the value of the corresponding pixel in the binary mask, which have values 0 and 1 for false and true, respectively. We now construct our

color-corrected image  $\mathbf{x}'$ , where the individual pixels have values

$$x'_{ijk} = x_{ijk} + m_{ij}(\bar{x}'_k - \bar{x}_k) \quad (6)$$

and  $\bar{x}'_k$  is the desired average value for the  $k$ -th channel of  $\mathbf{x}'$ . We found that choosing the CIELAB channels to have average values  $\bar{x}'_L = 170$ ,  $\bar{x}'_A = 100$ , and  $\bar{x}'_B = 160$  provides a suitable plant color. Note that in the definition of  $x'_{ijk}$ , one can assign pixel values that exceed the valid range  $[0, 255]$ . In this case, one must also constrain  $x'_{ijk}$  to the same range. In general, color-correction leads to a composite image where the plant has enhanced lightness, greenness, and yellowness. An example of a composite image with and without color-correction is shown in Figure 7.

For small-scale translation experiments, our input domain consists of 64 (optionally color-corrected) composite images. Larger-scale experiments use datasets composed of 512 composites. The method and parameters used to construct each dataset are summarized in Table 1.

### 3. Single-plant image translation results

In this section, we provide single-plant translation results for several GANs trained on the datasets described in Section 2. Training hyperparameters are consistent for all models, where we train for a total of 400 epochs and use a dynamic learning rate that begins to decay linearly to zero after the 200th epoch.

#### 3.1. Cropped lab dataset

Training a generator on the *Cropped Lab* dataset allows us to directly translate indoor plants to outdoor, without the intermediate step of creating composite images. This dataset contains 64 indoor and outdoor images of soybean (128 total), aged 10–40 days. Twenty additional indoor photos of soybean from the same age range unseen during the training process compose



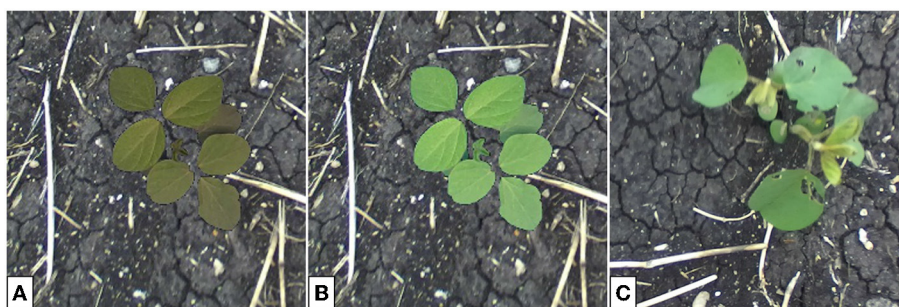


FIGURE 7

Composite image of a soybean plant without (A) and with (B) color-correction. The color-corrected image contains plant leaves that better match the outdoor domain (C). Images such as (A) or (B) can be used to provide the input domain  $X$  for generator training.



FIGURE 8

Soybean images sampled from the distribution  $Y$  (A), images sampled from the distribution  $X$  (B), translated images  $G(x)$  (C), images sampled from the distribution  $V$  (D), and translated images  $G(v)$  (E).

the distribution  $V$  for qualitative evaluation of the model. Figure 8 shows translation results for this generator.

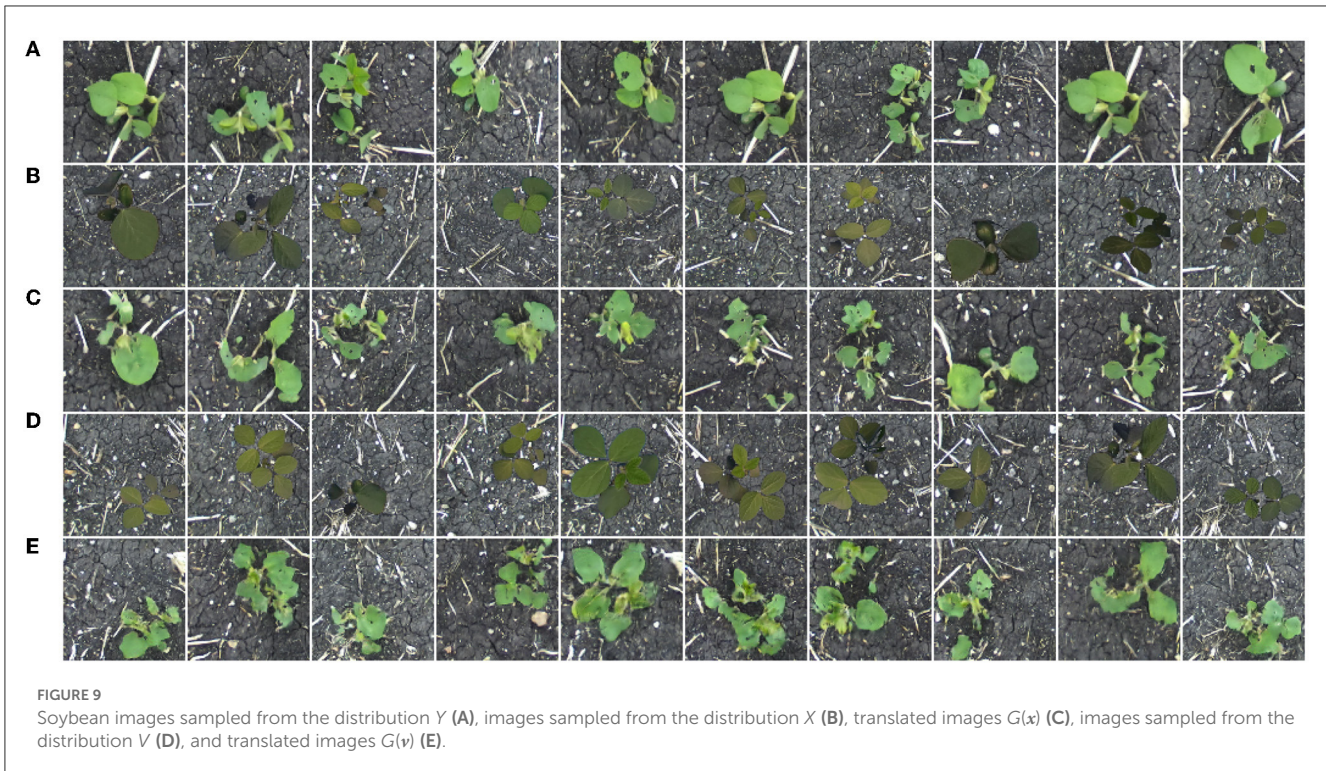
Referring to both the training and evaluation images, the generator appears to be capable of image translation between these two domains with little error. Details such as leaf color, leaf destruction, and soil are suitably added in each translated image. Leaf veins, which are greatly pronounced in some indoor images, are removed in the translated images, reflective of the images from the outdoor domain. Despite the ability to translate indoor plants to appear as true outdoor plants, the greatest drawback to a generator trained on these images is the inability to control the image background in the translated images. As discussed in Section 4, this is problematic for the construction of images with multiple plants, where one wants consistency in the background throughout the image.

### 3.2. Composites dataset

Training a generator on the *Composites* dataset allows us to translate uncorrected composite images to outdoor-appearing plants. Here, we expect that the background in the output image remains consistent with the input. This dataset contains 64 composite and outdoor images of soybean (128 total), aged 10–40 days. Twenty additional composite photos of soybean from the same age range unseen during the training process compose the distribution  $V$  for qualitative evaluation of the model. Figure 9 shows translation results for this generator.

Referring to both the training and evaluation images, the generator is capable of image translation between these two domains. Similar to the previous section, plant details such as leaf color, destruction, and veins are suitably added/removed in





the translated images. In general, the image backgrounds are consistent before and after translation. Small changes, such as the addition/removal of straw or pebbles in the background, are seen in the translated images. However, these details are sufficiently small so as not to have any negative effect during the construction of a synthetic multi-plant image.

### 3.3. Color-corrected composite datasets

Training a generator on the *Color-Corrected Composite* datasets allows us to translate color-corrected composite images to outdoor-appearing plants. Similar to the previous section, we expect that the background in the output image remains consistent with the input. The first generator in this section is trained on a dataset that contains 64 color-corrected composite and outdoor images of soybean (128 total), aged 10–40 days. Twenty additional color-corrected composite photos of soybean from the same age range unseen during the training process compose the distribution  $V$  for qualitative evaluation of the model. Figure 10 shows translation results for this generator. Additional translation results for generators trained on similar datasets of different species are shown in Appendix 2 (Supplementary material).

Referring to both the training and evaluation images, the generator is capable of image translation between these two domains. Similar to the previous section, plant details are suitably added or removed in the translated images, and the image background is consistent before and after translation. Additionally, little-to-no positional drift is seen between plants in the input and output images, indicating a model such as this could be used for multiple-plant image synthesis.

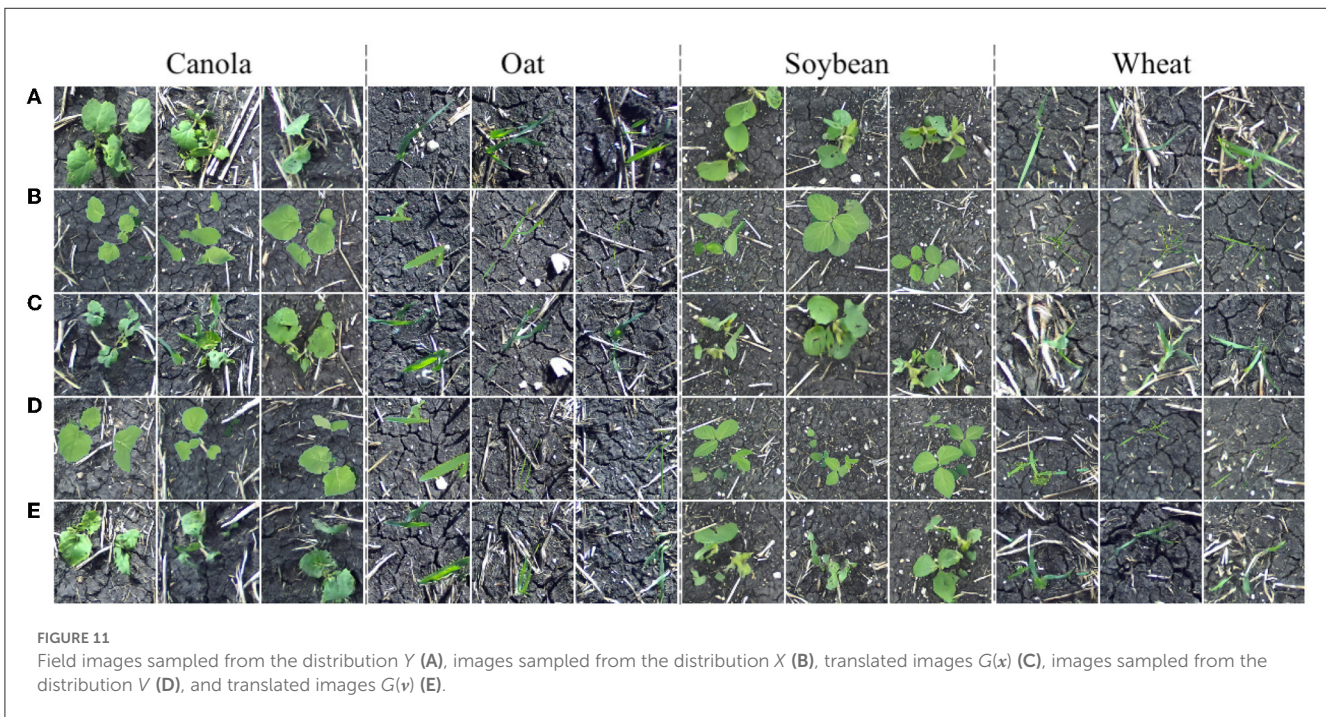
We now turn our focus to translating images of additional plant species. The *Color-Corrected Composites 2* training dataset contains 128 color-corrected composite and 128 field images of each of the four available species: canola, oat, soybean, and wheat (1,024 images total). Twenty additional color-corrected composites for each species compose the distribution  $V$  for qualitative evaluation of the model. Image translation results are given in Figure 11.

From the above translation results, it can be seen that our method is easily extendable to additional species. Translated plants have no noticeable positional drift and background appearance is consistent between the input and output. A model such as the one providing image translations in Figure 11 could be used to construct synthetic multiple-plant images in a field setting with multiple plant species while retaining the original bounding boxes to enable the production of object detection data.

## 4. Multiple-plant image translation results

The results in the previous section suggest that a generator trained to translate composite images would be suitable for multiple-plant image construction. Here, one expects to be able to place several synthetic plants within a real soil background to produce plausible outdoor images of multiple plants. Domain translation by the GAN should produce outdoor plant images with lightness similar to that seen typically in our outdoor data, as well as sufficient blurring of the plant relative to the real background. Using a single-plant translation generator, multi-plant images are constructed via the following algorithm. The algorithm makes use of existing full-scale soil images (see Figure 12A for an example) and indoor



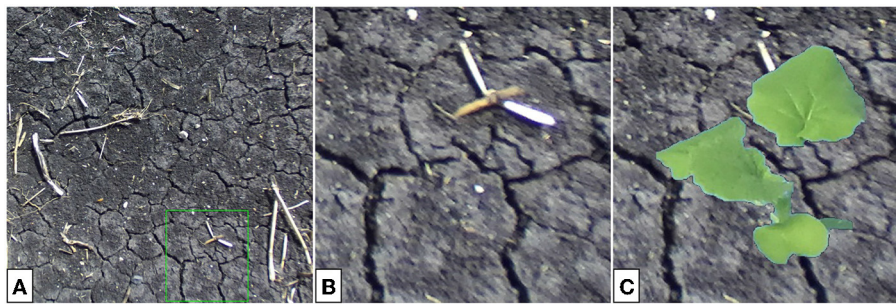


single-plant images with their binary masks (see [Figures 6A2, B2](#) for examples).

[Algorithm 1](#) effectively creates several composites, translates, and places them back into the full-scale image. Instead of selecting

composite backgrounds from a pre-defined set, small sections are randomly cropped from 20 larger images. [Figure 12](#) provides an example of composite background selection from a large-scale soil image.





**FIGURE 12**  
Full-scale soil image (A), cropped section of the image (B), and the color-corrected composite image to be passed to the generator (C). The cropping bounds are given by the green bounding box in (A).

```

1: randomly select a full-scale soil image
2: for number of plants do
3:   randomly select an indoor plant image
4:   select corresponding binary mask
5:   select a sub-section of the soil image as the
   composite background
6:   form a composite image using the image, mask,
   and background section
7:   if color-correct is true then
8:     color-correct the composite image
9:   end if
10:  translate composite image
11:  color-correct the translated image
12:  replace background sub-section with translated
   composite image
13: end for

```

**Algorithm 1.** Algorithm for constructing a synthetic outdoor multiple-plant image.

To improve generator performance when translating images with these new backgrounds, we construct an additional training dataset with 512 new color-corrected composite images and train a new CUT-GAN. Here, the composite backgrounds are randomly selected sub-sections of the large-scale backgrounds, as opposed to randomly selecting from the set of 128 small backgrounds used in the dataset *Color-Corrected Composites 2*. This training dataset uses the same 512 images as *Color-Corrected Composites 2* for the target domain  $Y$ . [Table 2](#) summarizes the parameters for constructing our new dataset.

In line 11 of [Algorithm 1](#), composite images are color-corrected following translation. This is a necessary step to help ensure a consistent background of our multi-plant images after translation. Background correction becomes especially useful if one chooses an image background that differs from our training data. If one were to train a model to detect plants in our synthetic images without correction, it is possible that the model learns to locate plants through inconsistencies in the background where the translated plant is placed. Background color-correction is used as an attempt to mitigate this risk. Additional processing can be done to improve

the joining of the translated composite with the background, but is not shown in this article.

The procedure for background correction follows similarly to that of color-correction for composite images. However, rather than only correcting the pixels belonging to the plant, we correct all pixels in the translated image by adding a uniform offset to all pixels of a given channel. This offset is found by calculating the difference in average values for each RGB channel in the composite and translated images, only for pixels in a small region outside the plant bounding box. In general, for background correction, we assume that the translated plant does not exceed the bounding box. However, small patches of green outside the bounding box (see [Figure 13B](#)) lead to little difference in the result.

To achieve background correction, three inputs are required: the  $(M \times N \times 3)$  composite single-plant image  $\mathbf{x}$ ,  $(M \times N \times 3)$  translated single-plant image  $\hat{\mathbf{y}}$ , and the  $(M \times N)$  binary mask  $\mathbf{m}$ . Example images are given in [Figure 13](#).

The offset  $d_k$  for the  $k$ -th channel of the corrected image  $\hat{\mathbf{y}}'$  is calculated as

$$d_k = \frac{\sum_{i=1}^M \sum_{j=1}^N (x_{ijk} - \hat{y}_{ijk}) m_{ij}}{\sum_{i=1}^M \sum_{j=1}^N m_{ij}}. \quad (7)$$

Equation (7) can equivalently be considered as the difference in weighted averages for the input and output images where the weight of all pixels within the bounding boxes is zero and one elsewhere. The pixels of  $\hat{\mathbf{y}}'$  are then given by

$$\hat{y}'_{ijk} = \hat{y}_{ijk} + d_k, \quad (8)$$

constrained to the range  $[0, 255]$ . [Figure 14](#) shows the effect of translating a single-plant image with and without applying background correction afterwards. [Figure 15](#) demonstrates the difference between multi-plant images with and without background correction.

In general, the translation capabilities of the generator are invariant to the position of plants in a given image. As a result, there is no restriction on placement, aside from avoiding overlapping plants. Our approach provides the option to place plants randomly, or by alignment into rows, as they would be in a production field.

TABLE 2 Parameters for the new training dataset.

Dataset name	$N_x$	$N_y$	Species	Age (days)	$N_{\text{backgrounds}}$	$S_{\text{min}}$	$S_{\text{max}}$
Color-corrected composites 3	512	512	All	Varies	512	0.50	0.85

The dataset contains each of the four available species; canola, oat, soybean, and wheat. These plants are aged 10–40, 0–365, 10–40, and 0–365 days, respectively. Each composite image possesses a unique background that comes from randomly cropping small sections from 20 large soil images.

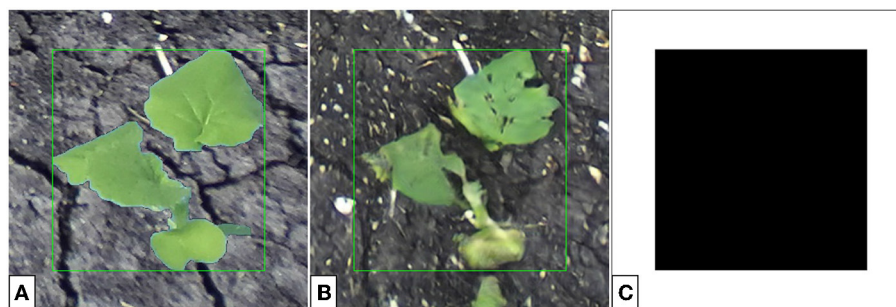


FIGURE 13

Color-corrected composite single-plant image (A), translated single-plant image (B), and the associated binary mask (C). The binary mask is false for all pixels within the bounding box and true for all other pixels.

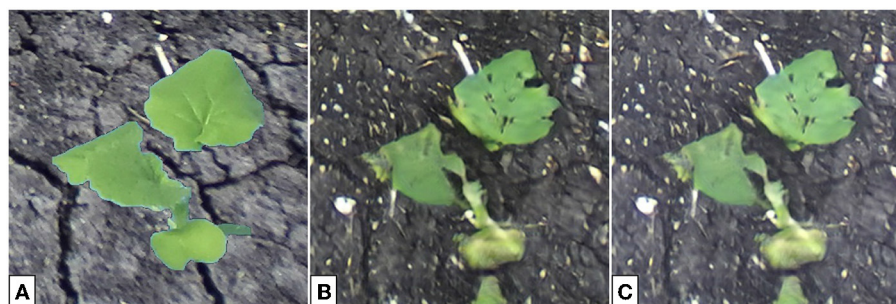


FIGURE 14

Single-plant images before (A) and after (B) translation and the corrected translated image (C). The soil color and lightness in (C) is more consistent with the input (A).

Additionally, the minimum and maximum scales of the plants relative to the background can be chosen by the user. Datasets can be made to contain plants of all species or individual plants can be selected. Examples of labeled synthetic multi-plant images are given in Figure 16. From the multi-plant images presented here, one can see that the position of each plant is maintained after translation, suggesting that such data could be useful for training a plant detection network.

## 5. Results of plant detection using augmented datasets

In general, object detection is a machine learning task that refers to the process of locating objects of interest within a particular image (Zhao et al., 2019). Models capable of object detection typically require training data consisting of a large number of images as well as the location and label of all objects within said images, usually given by bounding boxes. Such data

is often produced by manual annotation, a procedure that is both time- and cost-intensive, especially as the number of objects within an image increases (Guillaumin and Ferrari, 2012; Ayalew, 2020). As a result, object detection is an excellent test case of our image transformation methods.

As proof of concept that our synthetic multi-plant images can be beneficial for detection of real plants, we trained several YoloV5 (Redmon et al., 2016; Bochkovskiy et al., 2020) nano object detection models on various augmented datasets (Jocher et al., 2022). The nano is chosen in particular as it has the fewest number of trainable parameters in comparison to all other YoloV5 networks. This is most desirable for our training datasets which contain few classes and little variability in the data itself. In all cases the network is trained to locate canola, oat, soybean, and wheat. However, we are not attempting to find a solution to a classification problem, so these four species are grouped into a single class named *plant*. Three primary training dataset types are described below, and each includes the random placement of non-overlapping plants onto an image background. The size and location of the single plant



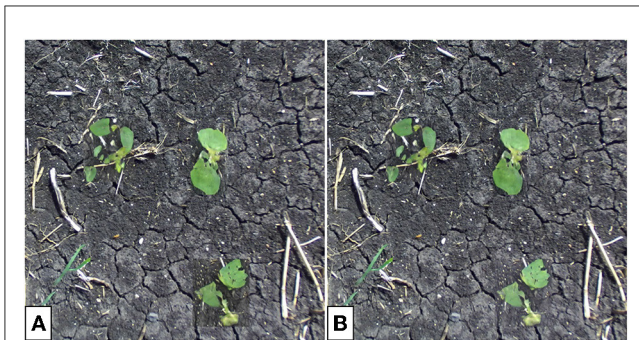


FIGURE 15

Translated multi-plant image without (A) and with (B) background correction. Note that the backgrounds of the translated sections of the image differ in color and lightness from the soil in the rest of the image. Background correction lessens this effect, especially seen for the soybean plant in the bottom-right.

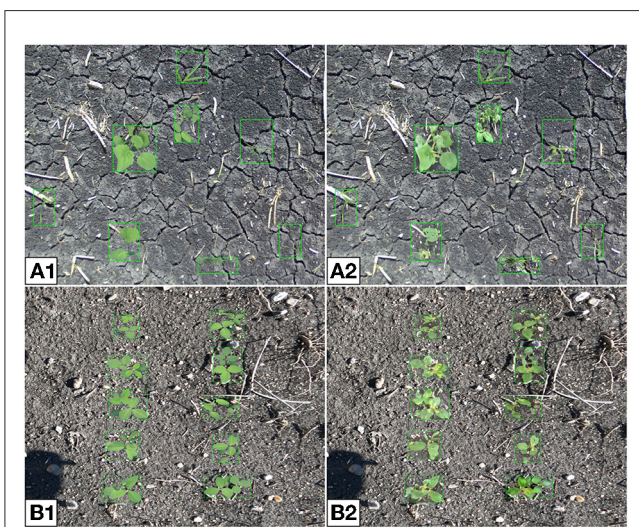


FIGURE 16

Composite (A) and synthetic (B) multi-plant images where plants are placed randomly (A1, B1) or ordered into two rows (A2, B2). The randomly placed plants are chosen to be any of the four available species, the ordered plants are all soybean.

image embedded within the background are used as the ground truth bounding boxes for our training data.

The first training dataset contains 80,000 blue screen images with color-corrected plants randomly placed throughout (Figure 17A). The dataset is split into training, validation, and testing sets with proportions 80% (64,000 images), 10% (8,000 images), and 10% (8,000 images), respectively. Note that no background replacement or GAN is used, so the differences between the training data and real field data are significant. Hence, we expect the performance of this network to be poor in general. For future references this dataset is referred to as the *Baseline* dataset.

The *Composite* dataset contains 80,000 multi-plant color-corrected composite images (see Figure 17B) with the same 80%

(64,000 images), 10% (8,000 images), and 10% (8,000 images) split as above. Here we include a real soil background, but no GAN is used to individually translate each plant. Since this dataset is more similar to real data than the baseline we expect to see improved performance on an evaluation set composed of real images.

The third dataset used to train the network consists of 80,000 multi-plant GAN images (see Figure 17C) split identically as above. This training data is created through the multi-plant GAN procedure previously described in this section. Here we expect network performance to be the greatest, since the training and target datasets are most similar. For future references this dataset is named *GAN*.

An additional dataset, known as the *Merged* dataset, is composed as the union of both the *Composite* and *GAN* datasets. As such, this training dataset consists of 160,000 total images for the network with half of the images including the usage of a GAN for plant translation.

With several plant detection models trained on various datasets, our models are evaluated on 253 additional real multi-plant images of canola and soy. These images were excluded during the training process and ground truth bounding boxes are determined by hand. The evaluation metrics used are precision, recall, and mean average precision (mAP). Before defining the metrics, one must first consider the intersection over union (IoU), given by Everingham et al. (2010):

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}, \quad (9)$$

where Area of Overlap refers to the area of overlap between the ground truth and predicted bounding boxes and Area of Union is the total area from joining the bounding boxes. Whether a bounding box prediction is considered to be successful is dependent on the IoU threshold, where we consider any prediction leading to an IoU greater than the threshold to be correct. Now, with IoU being used to determine if a model prediction is correct, the metrics precision and recall are defined by Everingham et al. (2010) as

$$\text{precision} = \frac{tp}{tp + fp}, \quad (10)$$

$$\text{recall} = \frac{tp}{tp + fn} \quad (11)$$

where  $tp$  denotes the number of true-positives,  $fp$  the false-positives, and  $fn$  the false-negatives. We interpolate precision over 101 recall values in the range [0.00, 1.00] in steps of 0.01. For notational simplicity, we define the set of recall values  $R = \{0.00, 0.01, \dots, 1.00\}$ . The interpolated precision is given by Everingham et al. (2010) as

$$\text{precision}_{\text{inter}}(r) = \max\{\text{precision}_{\tilde{r}; \tilde{r} \geq r}(\tilde{r})\}. \quad (12)$$

Finally, mAP is defined by Everingham et al. (2010) as

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^n \frac{1}{|R|} \sum_{r \in R} \text{precision}_{\text{inter}}(r), \quad (13)$$

where  $n$  denotes the number of classes and the outer sum leads to a mean precision for all classes. Note that since we group our species into a single class we have  $n = 1$  and the outer sum is presented here only for verbosity. The



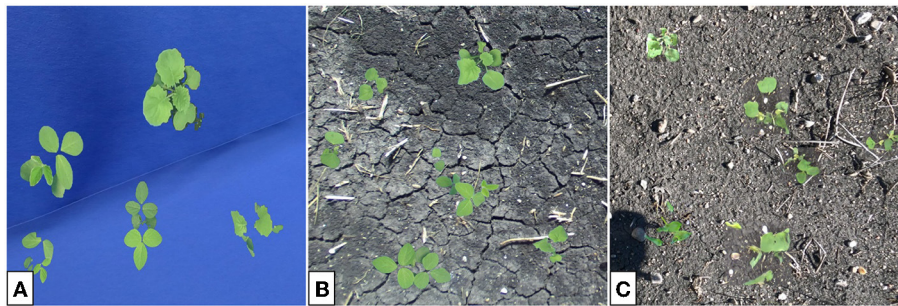


FIGURE 17

Sample images from the *Baseline* (A), *Composite* (B), and *GAN* (C) datasets used to train the YoloV5 nano networks. In all cases, plants are color-corrected and randomly placed into non-overlapping positions onto an image background. The *GAN* images receive the additional step of plant translation.

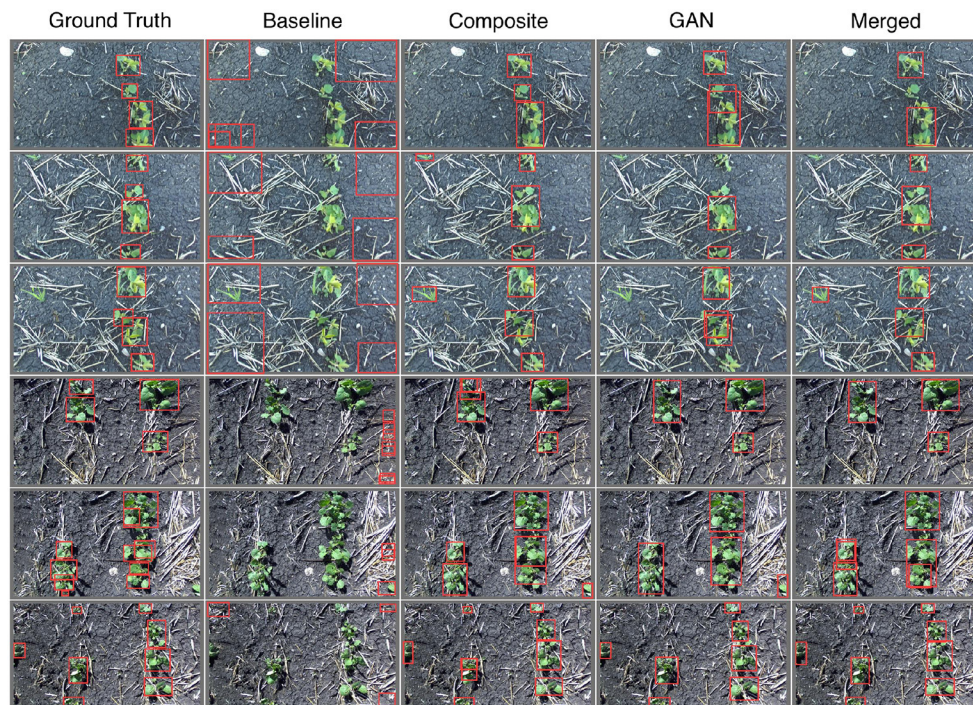


FIGURE 18

Sample images labeled by YoloV5 nano object detection models trained on various augmented datasets (see labels). Note that the ground truth bounding boxes are determined by hand. The top three image rows contain only soybean plants, the bottom three contain only canola.

notation  $mAP@0.5$  denotes the evaluation of  $mAP$  using an IoU threshold of 0.5, consistent with the Pascal VOC evaluation metric (Everingham et al., 2010). Alternatively,  $mAP@0.5:0.95$  denotes the average  $mAP$  value for all IoU thresholds in the range [0.50, 0.95], in steps of 0.05, which is identical to the evaluation metric for the COCO dataset challenge (Lin et al., 2014). Sampled visual results for the performance of each network are given in Figure 18, numerical results are provided by Table 3, including both the  $mAP@0.5$  and  $mAP@0.5:0.95$  metrics.

From both Figure 18 and Table 3 it is clear that the baseline model shows no capacity to detect plants. This is expected as the training data is vastly different from the real images through which our models are evaluated. Both the *Composite*- and *GAN*-trained models show good ability to locate plants in the evaluation images, however the *GAN*-trained model performs better according to both the  $mAP@0.5$  and  $mAP@0.5:0.95$  metrics. The combination of the *Composite* and *GAN* datasets led to the greatest performing network in terms of our metrics. However, this could come as a result

TABLE 3 Precision, recall, and mAP metrics when evaluating our four YoloV5 nano models on 253 real images of canola and soybean.

Training dataset	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Baseline	0.030	0.099	0.010	0.002
Composite	0.597	0.468	0.454	0.202
GAN	0.647	0.459	0.479	0.209
Merged	0.673	0.500	0.528	0.240

The training dataset denotes the images used to train the network for which we are evaluating, not the set from which the evaluation images are taken.

of being exposed to twice the number of training images. In general, the Yolo models trained on the *Composite*, *GAN*, and *Merged* datasets all appear capable of plant detection on the provided images.

## 6. Conclusion

The contribution of this work is an image translation process through which one can produce artificial images in field settings, using images of plants taken in an indoor lab. The method is easily extendable to new plant species and settings, provided there exists sufficient real data to train the underlying GAN. The construction of one's own augmented datasets enables further training of neural networks, with applications to in-field plant detection and classification. More importantly, this work is a first step in demonstrating that plants grown in growth chambers under precise and fully-controlled conditions can be used to easily generate large amounts of labeled data for developing machine learning models that operate in outdoor environments. This work has the potential to significantly improve and accelerate the model development process for machine learning applications in agriculture.

Future work will consist of mimicking outdoor lighting conditions via controllable led-based growth-chamber lights, leading to more variety in the input data and hopefully more realistic synthetic data. Additional work will focus on using the approach presented here to develop in-field plant-classification models, as well as developing outdoor datasets and machine learning models for other problem domains (such as disease detection). This will include an investigation into whether classification tasks require their own species-specific GAN for the image translation process.

## Data availability statement

The datasets presented in this study can be found in online repositories. All the data for our article is available from the online portal we have created to share our data. This portal is hosted by the Digital Research Alliance of Canada, and access can be obtained by contacting the authors.

## Author contributions

AK: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, review, and editing. PS: data curation, formal analysis, investigation, software, validation, visualization, and writing—original draft. CH: conceptualization, formal analysis, funding acquisition, methodology, project administration, resources, supervision, writing—original draft, review, and editing. MB: data curation, formal analysis, methodology, supervision, and writing—review and editing. CB: formal analysis, funding acquisition, project administration, resources, supervision, and writing—review and editing. All authors contributed to the article and approved the submitted version.

## Funding

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant program (Nos. RGPIN-2018-04088 and RGPIN-2020-06191), Compute Canada (now Digital Research Alliance of Canada) Resources for Research Groups competition (No. 1679), Western Economic Diversification Canada (No. 15453), and the Mitacs Accelerate Grant program (No. IT14120).

## Acknowledgments

The authors would like to thank Ezzat Ibrahim for establishing the Dr. Ezzat A. Ibrahim GPU Educational Lab at the University of Winnipeg, which provided the computing resources needed for this work.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frai.2023.1200977/full#supplementary-material>



## References

- Ayalew, T. W. (2020). *Unsupervised domain adaptation for object counting* (Doctoral dissertation). University of Saskatchewan, Saskatchewan, SK, Canada.
- Bah, M. D., Hafiane, A., and Canals, R. (2018). Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sensing* 10. doi: 10.3390/rs10111690
- Barth, R., Hemming, J., and van Henten, E. J. (2018). Improved part segmentation performance by optimising realism of synthetic images using cycle generative adversarial networks. *arXiv preprint arXiv:1803.06301*. doi: 10.48550/arXiv.1803.06301
- Beck, M. A., Liu, C.-Y., Bidinosti, C. P., Henry, C. J., Godee, C. M., and Ajmani, M. (2020). An embedded system for the automated generation of labeled plant images to enable machine learning applications in agriculture. *PLoS ONE* 15, e0243923. doi: 10.1371/journal.pone.0243923
- Beck, M. A., Liu, C.-Y., Bidinosti, C. P., Henry, C. J., Godee, C. M., and Ajmani, M. (2021). Presenting an extensive lab- and field-image dataset of crops and weeds for computer vision tasks in agriculture. *arXiv preprint arXiv:2108.05789*. doi: 10.48550/arXiv.2108.05789
- Binch, A., and Fox, C. (2017). Controlled comparison of machine vision algorithms for Rumex and Urtica detection in Grassland. *Comput. Electron. Agric.* 140, 123–138. doi: 10.1016/j.compag.2017.05.018
- Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). YOLOV4: optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. doi: 10.48550/arXiv.2004.10934
- Bosilj, P., Duckett, T., and Cielniak, G. (2018). Analysis of morphology-based features for classification of crop and weeds in precision agriculture. *IEEE Robot. Automat. Lett.* 3, 2950–2956. doi: 10.1109/LRA.2018.2848305
- Cap, Q. H., Uga, H., Kagiwada, S., and Iyatomi, H. (2022). Leafgan: An effective data augmentation method for practical plant disease diagnosis. *IEEE Trans. Automat. Sci. Eng.* 19, 1258–1267. doi: 10.1109/TASE.2020.3041499
- Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* 88, 303–338. doi: 10.1007/s11263-009-0275-4
- Fahlgren, N., Gehan, M. A., and Baxter, I. (2015). Lights, camera, action: high-throughput plant phenotyping is ready for a close-up. *Curr. Opin. Plant Biol.* 24, 93–99. doi: 10.1016/j.pbi.2015.02.006
- Gehan, M. A., and Kellogg, E. A. (2017). High-throughput phenotyping. *Am. J. Bot.* 104, 505–508. doi: 10.3732/ajb.1700044
- Giuffrida, M. V., Chen, F., Scharr, H., and Tsafaris, S. A. (2018). Citizen crowds and experts: observer variability in image-based plant phenotyping. *Plant Methods* 14, 12. doi: 10.1186/s13007-018-0278-7
- Giuffrida, M. V., Scharr, H., and Tsafaris, S. A. (2017). “Arigan: synthetic arabidopsis plants using generative adversarial network,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (Venice), 2064–2071.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks.
- Guillaumin, M., and Ferrari, V. (2012). “Large-scale knowledge transfer for object localization in imagenet,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Providence, RI), 3202–3209.
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*. doi: 10.48550/arXiv.1611.07004
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., et al. (2022). *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*.
- Kuznichov, D., Zvirin, A., Honen, Y., and Kimmel, R. (2019). “Data augmentation for leaf segmentation and counting tasks in rosette plants,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Long Beach, CA), 2580–2589.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539
- Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. (2018). Machine learning in agriculture: a review. *Sensors* 18. doi: 10.3390/s18082674
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., et al. (2014). Microsoft COCO: common objects in context. *arXiv preprint arXiv:1405.0312*. doi: 10.48550/arXiv.1405.0312
- Lobet, G. (2017). Image analysis in plant sciences: publish then perish. *Trends Plant Sci.* 22, 559–566. doi: 10.1016/j.tplants.2017.05.002
- Lu, Y., Chen, D., Olaniji, E., and Huang, Y. (2022). Generative adversarial networks (GANs) for image augmentation in agriculture: a systematic review. *Comput. Electron. Agric.* 200, 107208. doi: 10.1016/j.compag.2022.107208
- Madsen, S. L., Dyrmann, M., Jorgensen, R. N., and Karstoft, H. (2019). Generating artificial images of plant seedlings using generative adversarial networks. *Biosyst. Eng.* 187, 147–159. doi: 10.1016/j.biosystemseng.2019.09.005
- Park, T., Efros, A. A., Zhang, R., and Zhu, J.-Y. (2020). Contrastive learning for unpaired image-to-image translation. *arXiv preprint arXiv:2007.15651*.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). “You only look once: unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV), 779–788.
- Shakoor, N., Lee, S., and Mockler, T. C. (2017). High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. *Curr. Opin. Plant Biol.* 38, 184–192. doi: 10.1016/j.pbi.2017.05.006
- Singh, A., Ganapathysubramanian, B., Singh, A. K., and Sarkar, S. (2016). Machine learning for high-throughput stress phenotyping in plants. *Trends Plant Sci.* 21, 110–124. doi: 10.1016/j.tplants.2015.10.015
- Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., and Bennett, M. (2017). Plant phenomics, from sensors to knowledge. *Curr. Biol.* 27, R770–R783. doi: 10.1016/j.cub.2017.05.055
- Waldchen, J., Rzanny, M., Seeland, M., and Mader, P. (2018). Automated plant species identification-trends and future directions. *PLoS Comput. Biol.* 14, e1005993. doi: 10.1371/journal.pcbi.1005993
- Zeng, Q., Ma, X., Cheng, B., Zhou, E., and Pang, W. (2020). GANs-based data augmentation for citrus disease severity detection using deep learning. *IEEE Access* 8, 172882–172891. doi: 10.1109/ACCESS.2020.3025196
- Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object detection with deep learning: a review. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865
- Zhu, F., He, M., and Zheng, Z. (2020). Data augmentation using improved CDCGAN for plant vigor rating. *Comput. Electron. Agric.* 175, 105603. doi: 10.1016/j.compag.2020.105603
- Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*. doi: 10.48550/arXiv.1703.10593
- Zhu, Y., Aoun, M., Krijn, M., and Vanschoren, J. (2018). “Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants,” in *British Machine Vision Conference* (Newcastle).