

High-Resolution UAV Image Generation for Sorghum Panicle Detection

Enyu Cai, Zhankun Luo, Sriram Baireddy, Jiaqi Guo, Changye Yang, Edward J. Delp
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

{cenyu, luo333, sbairedd, guo498, yang853, ace}@ecn.purdue.edu

Abstract

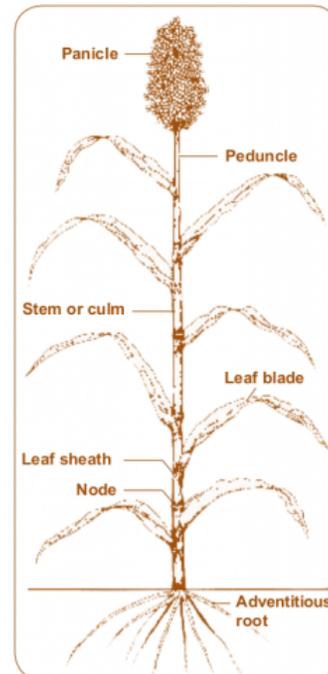
The number of panicles (or heads) of Sorghum plants is an important phenotypic trait for plant development and grain yield estimation. The use of Unmanned Aerial Vehicles (UAVs) enables the capability of collecting and analyzing Sorghum images on a large scale. Deep learning can provide methods for estimating phenotypic traits from UAV images but requires a large amount of labeled data. The lack of training data due to the labor-intensive ground truthing of UAV images causes a major bottleneck in developing methods for Sorghum panicle detection and counting. In this paper, we present an approach that uses synthetic training images from generative adversarial networks (GANs) for data augmentation to enhance the performance of Sorghum panicle detection and counting. Our method can generate synthetic high-resolution UAV RGB images with panicle labels by using image-to-image translation GANs with a limited ground truth dataset of real UAV RGB images. The results show the improvements in panicle detection and counting using our data augmentation approach.

1. Introduction

Crop-based biofuels, an environmentally sustainable energy resource derived from plant matter, can help reduce greenhouse gas emissions and dependency on exhaustible resources [8, 28]. In recent studies, Sorghum (*Sorghum bicolor* (L.) Moench) has been found to be a very good potential biofuel [28, 37, 44]. Biofuel from Sorghum is less of a concern than biofuel from main food crops such as corn and sugar cane since the production of biofuel from these food crops could lead to supply shortages in food production [37]. A Sorghum panicle is a cluster of flowers at the top of the plant as shown in Figure 1. The “panicle” (or “head”) of Sorghum is one of the most important features of the plant related to grain yield, plant growth, and management [7, 32]. Detecting and counting the panicles can



(a) A Sorghum field (from [35]). The panicle is the cluster structure at the top of the Sorghum plants.



(b) Anatomy of Sorghum (from [33]).

Figure 1. Sorghum panicles.

provide information such as flowering dates to help plant breeders select the proper type of crops [4, 7, 24, 32].

Plant phenotyping is the methodology of linking a plant's observable characteristics to genetic information by estimating the phenotypic traits of plants such as height, shape, quantity, and color [45]. Traditional field-based phenotyping usually consists of walking through the field and manually estimating the plant traits using destructive methods which are not suitable on a large scale [19]. To address the limitations of traditional phenotyping, modern high-throughput phenotyping [5, 9, 20, 26] uses remote sensing with Unmanned Aerial Vehicles (UAVs) to collect data. The use of UAVs has dramatically improved phenotyping in speed, scale, and quality in a non-destructive manner by flying at low altitudes and collecting data rapidly. UAVs equipped with RGB sensors provide images with high spatial and temporal resolution for plant trait analysis. Typically, images acquired by UAVs are orthorectified with accurate location properties into orthomosaic [12] for data processing. This allows plant traits to be accurately measured.

Deep convolutional neural networks (CNN) [22] have achieved outstanding results in image classification, object detection and segmentation [13, 39, 43]. The high accuracy for identifying objects in images with complicated backgrounds makes CNNs attractive for estimating phenotypic traits. For deep neural networks, a large quantity of training data is often required to prevent overfitting. Labeling a large amount of training data is time-consuming and tedious due to the high spatial resolution and density of panicles in plant images. The lack of training data is a major bottleneck that affects the performance of the deep neural network in plant trait estimation. Data augmentation [3] is a technique to reduce overfitting by creating more training data samples such as rotation, flip, and crop on the existing dataset without additional ground truth. Adding synthetic images has become popular for data augmentation since the introduction of generative adversarial networks (GANs) [11], which can generate realistic images. GAN-based methods have a positive impact for enhancing the performance for classification tasks for plant images [2, 27]. Most GAN-based approaches focus on classification by generating images that contain a single instance of the plant. In plant trait estimation such as panicle detection, there are multiple instances of plants in an image, so the ability to generate images containing multiple objects with labels is necessary.

In this paper, we propose a method for generating high-resolution synthetic Sorghum UAV RGB images with bounding box labels using two image-to-image translation GANs. We create a label map (or label images) that contains bounding boxes where we want the panicle to be located in the synthetic image generated by the GAN. We shall denote each bounding box in the label map as a



(a) A real UAV RGB image of a Sorghum field.



(b) A synthetic UAV RGB image of a Sorghum field generated from our GAN.

Figure 2. An example of real and synthetic UAV RGB images of a Sorghum field.

”mask”. The GAN then generates a synthetic image with the panicle located as proscribed in the label map. We utilize a small amount of ground truth real images to train the image-to-image translation GAN with the label map. The synthetic images are combined with real images to train the object detection network. We compare the performance of the two image-to-image translation GANs. The results show performance improvement in both panicle detection and counting. Figure 2 shows an example of real and synthetic UAV RGB images of Sorghum.

2. Related Work

Generative Adversarial Network (GAN). The basic idea of a GAN [11] is to generate images that are indistinguishable from real images. A GAN consists of two models: the generator model for image generation and the discriminator model for classification of real and synthetic images. The generator often uses an inside-out auto-encoder [14] structure that takes random noise as input and generates synthetic images. The discriminator uses a CNN and multilayer perceptron (MLP) [10] as a classifier to distinguish the real and synthetic data. During the training of the GAN, the objective of the discriminator is to maximize the cost value, while the objective of the generator is to minimize the cost value. Developing a stable training approach is the main challenge in recent GAN research [16]. Since the GAN has two models (Generator and Discriminator), it is difficult to train both models simultaneously while ensuring convergence. There are a variety of GAN variants that focus on improving training stability. For example, Arjovsky *et al.* proposed the Wasserstein GAN [1] with a new loss function based on the Earth Mover’s Distance [41] to enhance training stability. Radford *et al.* developed a new model DCGAN [36] using a novel CNN architecture for both the generator and discriminator. The Conditional GAN (cGAN) [30] is a variant of a GAN in which auxiliary inputs, such as labels or text, are added as inputs to both the generator and discriminator so that the output can be controlled by the additional information. The Conditional GAN has wide applications such as text-to-image translation [38], image-to-image translation [25], and style transfer [18].

Data Augmentation Using Synthetic Images. Generating synthetic images from GANs for data augmentation is widely used to enhance the performance of deep neural networks for various tasks where there is insufficient training data [3]. The synthetic data is combined with real ground truth image to train a target network. Most GAN-based data augmentation methods are used to improve classification that identifies the single object from images. For example, in [2], Bi *et al.* developed a method using images generated by a Wasserstein GAN variant to improve plant disease classification. In [27], Madsen *et al.* improved plant seeds classification by using synthetic plant seed images from GAN. The Conditional GAN is often used in situations that require additional annotations for the synthetic images, such as data augmentation for detection and segmentation tasks. For example, in [29], Milz *et al.* proposed a conditional GAN to generate images with ground truth data for aerial object detection and segmentation. In [42], Sandfort *et al.* improved CT segmentation by using synthetic images from CycleGAN [48].

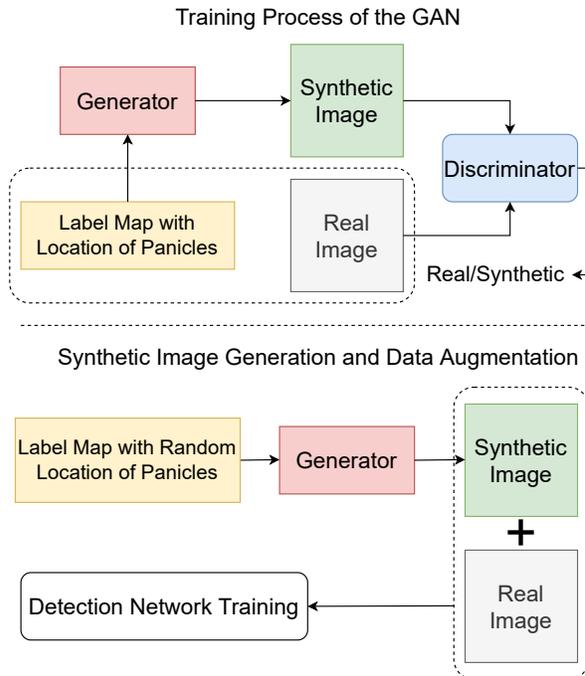


Figure 3. Block diagram of our proposed approach.

3. Proposed Approach

Our method consists of two parts: training a conditional GAN with a small amount of ground truth UAV RGB images (label maps) of Sorghum and then generating synthetic images with random locations of the panicles which we use for data augmentation. Figure 3 illustrates our proposed approach. In the top part of Figure 3, label maps and images are used for training the GAN. The label map is a grayscale mask acquired by converting existing ground truth bounding box labels. In the bottom part of Figure 3, we use random locations of the panicles (random label map) as an input to the GAN to generate synthetic UAV RGB images. The synthetic images are then added to the real ground truth image dataset to form the training data we will use to train the panicle detection network. The requirements for the GAN are that it must be capable of generating high-resolution images constrained by the inputs. We choose two image-to-image translation GANs because the real ground image bounding box labels can be converted directly into label masks. The two GANs we use are pix2pixHD [46] and SPADE [31] because of their reported performance on public datasets such as ADE20K [47] and Cityscapes [6].

3.1. pix2pixHD

The pix2pixHD [46] architecture is an image-to-image translation GAN that can generate high-resolution images from label map inputs. The concept of pix2pixHD comes

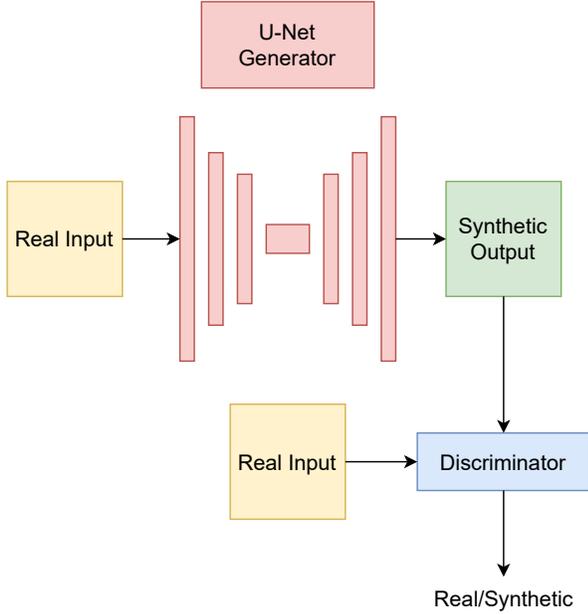


Figure 4. The pix2pix structure.

from pix2pix [15], a conditional GAN for image-to-image translation. The original pix2pix has the structure shown in Figure 4. pix2pix comprises of a generator G and a discriminator D . It has the following objective function:

$$\mathcal{L}_{\text{GAN}}(G, D) \equiv \mathbb{E}_{(\mathbf{s}, \mathbf{x})}[\log D(\mathbf{s}, \mathbf{x})] + \mathbb{E}_{\mathbf{s}}[\log(1 - D(\mathbf{s}, G(\mathbf{s})))] \quad (1)$$

where (s_i, x_i) is the i -th sample pair of a label and its corresponding nature image. We denote $\mathbb{E}_{(\mathbf{s}, \mathbf{x})} \triangleq \mathbb{E}_{(\mathbf{s}, \mathbf{x})} \sim p_{\text{data}}(\mathbf{s}, \mathbf{x})$. The generator and discriminator of pix2pix are derived from the DCGAN [36] which uses a CNN as its main structure. The generator of pix2pix uses skip connections from U-Net [40] to share information between input and output. The discriminator of pix2pix adapts the PatchGAN [15] structure. The PatchGAN is a patch-based CNN that requires fewer parameters to train compared to fully connected structures of other discriminators.

pix2pix can only generate images with a maximum resolution of 256×256 pixels. If we train pix2pix with images larger than 256×256 pixels, the training process becomes unstable and the image quality deteriorates. pix2pixHD [46] addresses the limitation of pix2pix by introducing multi-scale generators and discriminators. The structure of pix2pixHD is shown in Figure 5. It has the following objective function:

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right) \quad (2)$$

where λ controls the importance of a GAN loss

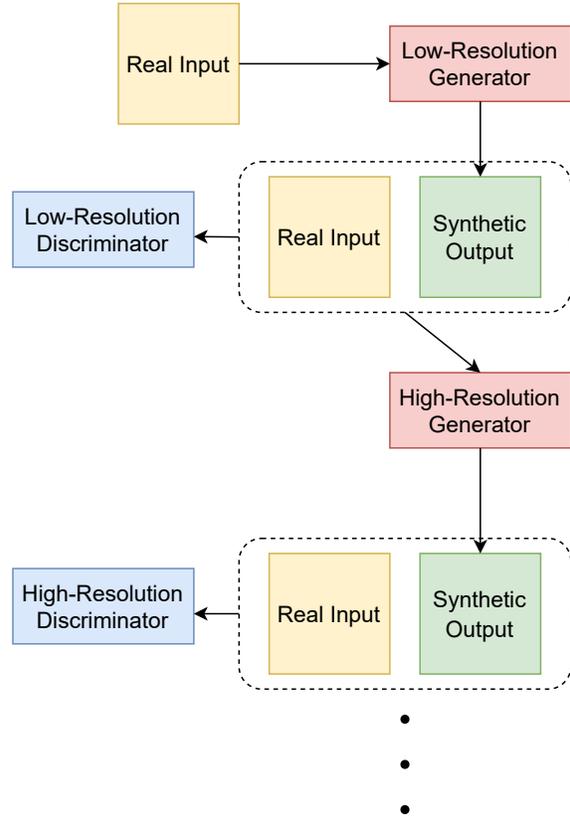


Figure 5. The pix2pixHD structure.

$\mathcal{L}_{\text{GAN}}(G, D_k)$ and a feature matching loss $\mathcal{L}_{\text{FM}}(G, D_k)$. The k -th feature matching loss $\mathcal{L}_{\text{FM}}(G, D_k)$ for the discriminator is incorporated in pix2pixHD to stabilize training:

$$\mathcal{L}_{\text{FM}}(G, D_k) \equiv \mathbb{E}_{(\mathbf{s}, \mathbf{x})} \sum_{i=1}^T \frac{1}{N_i} \|D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))\|_1 \quad (3)$$

where $D_k^{(i)}$ is the feature extractor of discriminator D_k for i -th layer, N_i denotes the number of elements in the i -th layer and T denotes the total number of layers.

The multiscale generators and discriminators form pyramid structures for synthesizing images with different resolutions. The generator structures of pix2pixHD consist of a global generator G_1 and a local enhancer network G_2 . During training, G_1 is first trained on low-resolution images. G_2 is appended to the last layer of G_1 and trained jointly on high-resolution images. The discriminator structures of pix2pixHD are paired with multiscale generators for low-resolution and high-resolution images. The input of its discriminator is real images and synthetic images that are downsampled at multiple scales that match the pyramid

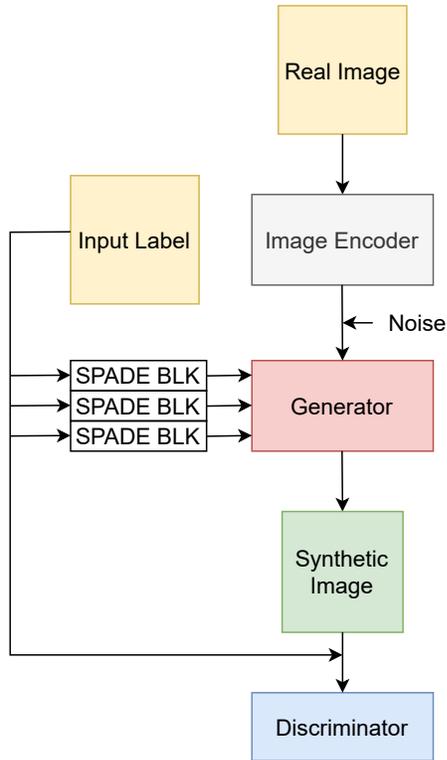


Figure 6. The SPADE structure.

structures. The high-resolution discriminator can encourage generating images with finer details. The low-resolution discriminator guides the generator to generate globally consistent images. pix2pixHD also uses an optional instance map as an auxiliary input. Each individual object inside the instance map has a unique ID. The instance map provides the object boundaries of the same class. However, for our application creating panicle segmentation masks for small objects is very time-consuming, which conflicts with our goal of saving time in ground truthing. We decide not to use instance maps for generating Sorghum images.

3.2. SPADE

The regular normalization layers from the generator of pix2pixHD tend to lose semantic information. The diminished semantic information affects the quality of the synthetic images. Spatially-adaptive denormalization (SPADE) [31] preserves the semantic information compared to regular normalization by incorporating residual blocks with SPADE [31] layers into the up-sampling part of the generators as shown in Figure 6. The SPADE layer consists of a convolutional layer and two modulation layers. The convolutional layer during training uses ground truth label mask as input. Two modulation layers take the output of

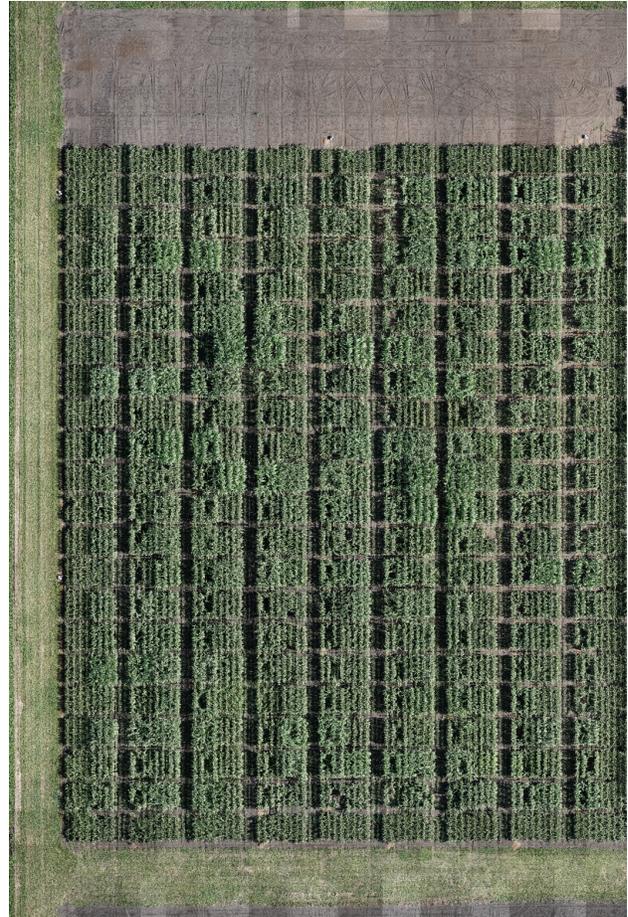


Figure 7. An orthomosaic of a Sorghum field. The dataset used for the experiments is cropped from a set of orthomosaics.

the convolutional layer and output two modulation parameters γ and β . The γ and β are multiplied and added to the normalized activation as output.

The generator of SPADE discards the encoder structures from pix2pixHD since the label maps are directly fed into the residual blocks within the generators. A new image encoder is used to encode the real image into random vectors before sending it into the up-sampling generator. The discriminator of SPADE follows the multi-scale discriminators of pix2pixHD with SPADE layers as normalization. The discriminator takes the concatenation of the input ground truth mask and the synthetic images and classifies whether the generated image is synthetic.

Compared to pix2pixHD, SPADE has better performance on the public datasets such as ADE20K [47] and Cityscapes [6] due to the added SPADE blocks. All of the public datasets have multiple classes and well-defined segmentation masks that contain a large amount of semantic information. We only have one class in our dataset (panicles) with minimal semantic information so the added SPADE

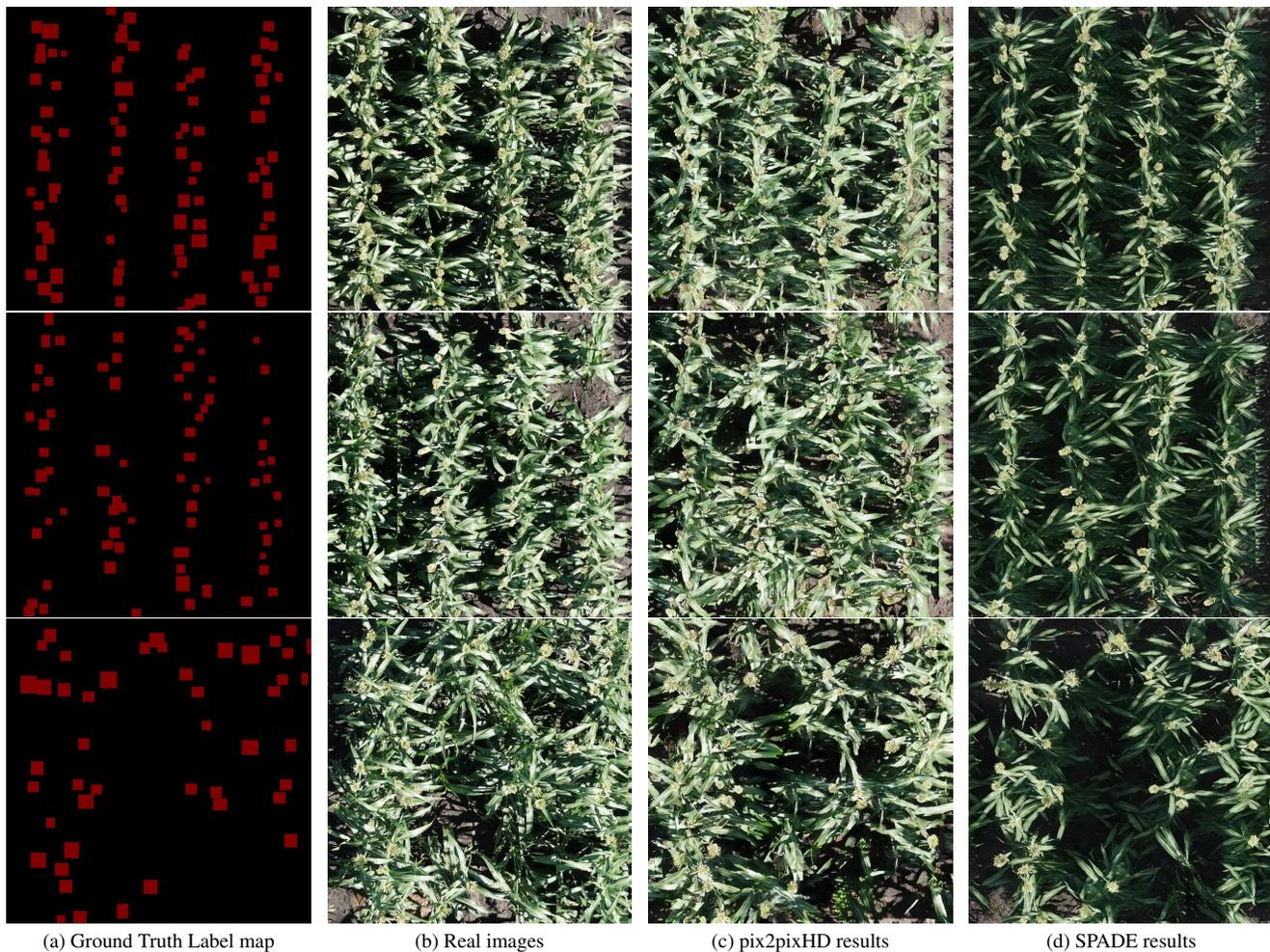


Figure 8. Real panicle images and synthetic panicle images generated using pix2pixHD and SPADE.

blocks for preserving semantic information might not be beneficial. We use both methods, pix2pixHD and SPADE, and compare their performance for synthetic panicle image generation.

4. Experiments

4.1. Dataset

The real image dataset consists of a set of cropped images from 4 orthomosaics of Sorghum field in acquired in 2020 as shown in Figure 7. The image is acquired by RGB cameras mounted on UAVs flying at an altitude of 20m. The spatial resolution of the orthomosaics is 0.25cm/pixel. There is a total of 4 orthomosaics from multiple dates that represent different growing stages of Sorghum. The resolution of the cropped image is 1024×1024 pixels. Each of the cropped images contains 4 row segments of Sorghum plants in the vertical direction. Each panicle in the images is manually labeled with a bounding box. The bounding box labels

are converted to the ground truth label maps for training the GANs. The label map is a single channel grayscale image that has the same resolution as the image. The conversion has the following equation:

$$p_{map} = \begin{cases} 1, & p_{map} \in R \\ 0, & p_{map} \notin R \end{cases} \quad (4)$$

where p_{map} is the pixel in the label map. R is the region inside the bounding box. The pixel in the label map has a value of 1 if the pixel is inside the bounding box. The rest of the pixel has a value of 0. In total, we have 500 manually labeled, cropped images from 4 orthomosaics as our experimental dataset. We use 400 images for training for the GANs and 100 images for testing the panicle detection and counting system. The number of cropped images from each orthomosaic is evenly distributed.

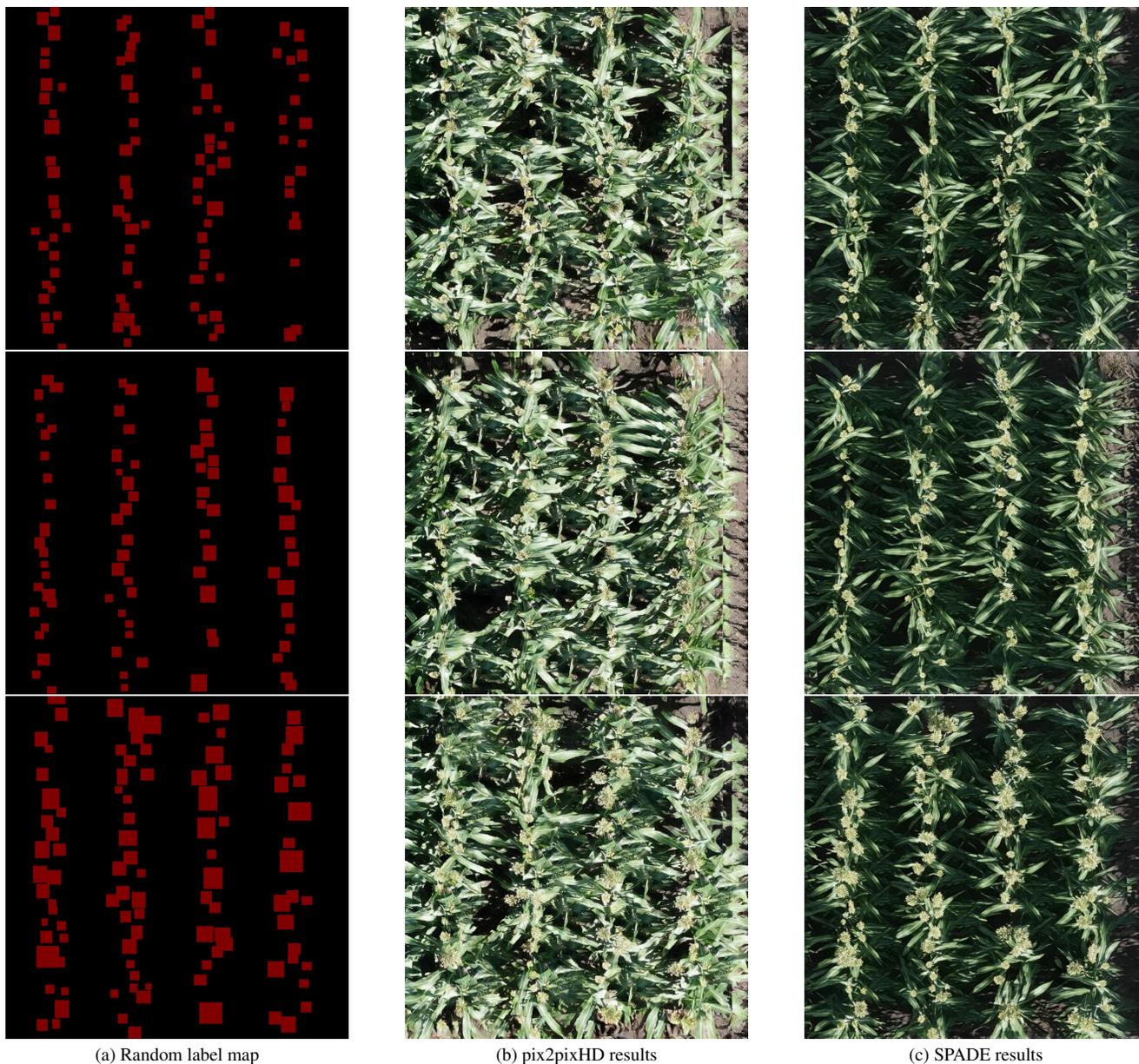


Figure 9. Synthetic panicle images generated using pix2pixHD and SPADE with random input labels.

4.2. Synthetic Panicle Generation

We train pix2pixHD and SPADE with the training dataset described above. Both methods are trained on a single NVIDIA TITAN RTX with 24GB memory and 200 epochs in native resolution (1024×1024 pixels) without any resizing or cropping. Adam optimization [21] is used for both pix2pixHD and SPADE with an initial learning rate of 0.0002. The results are shown in Figure 8. The panicles generated by SPADE are more realistic than those generated by the pix2pixHD. The panicle edges from SPADE are clearer while the panicle edges of pix2pixHD are blurred.

However, the pix2pixHD has a more natural background that resembles real Sorghum images. The synthetic background, e.g. the highlights on leaves and the overall brightness of SPADE, is too dark. The straight line on the right side of the synthetic images is learned from the real images created by the process of orthorectification. Overall, the images generated by pix2pixHD are more realistic.

4.3. Panicle Detection

We generated 1000 synthetic images each from pix2pixHD and SPADE using random label maps (a total

of 2000 GAN generated images). Figure 9 shows some of the examples of the synthetic images. The random label maps are constrained in location and size of the bounding boxes such that the overall distribution is identical to real Sorghum images in our dataset. A small overlap is allowed for each bounding box mask. In total, we have 1400 images for training our panicle detector from each of the GANs. This is 400 real ground truth images plus 1000 synthetic images for the particular GAN.

We use YOLOv5 [17] as our object detection network for panicles. We use the mean average precision (mAP) with Intersection over Union (IoU) from 0.5 to 0.95 (COCO mAP [23]) as the metric for detection, and Root Mean Squared Error (RMSE) [34], Mean Absolute Error (MAE) [34] and Absolute Percent Error (MAPE) [34] for counting:

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{AP} = \sum_k (R_k - R_{k-1}) P_k \quad (7)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|e_i|}{C_i} \quad (8)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |e_i| \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N |e_i|^2} \quad (10)$$

where k refers to the k -th threshold for precision and recall. C_i is the ground truth count in the i -th image. e_i is the difference between C_i and the estimated count. N is the number of image samples. In our case, the mAP is equal to AP because we only have one class. We evaluate the performance with the rest of 100 real ground truth images not used for training. The results are shown in Table 1. Compared to training only on 400 real ground truth images, both GAN-based data augmentation methods achieve better mAP. For the counting metrics, the model trained on the augmented images (real plus synthetic) also performs better. Despite the different image styles generated by pix2pixHD and SPADE, their data augmentation performances are similar. The synthetic images of pix2pixHD have slightly better performance for the counting metrics due to the more realistic background. We found that for the pix2pixHD the weights tend to diminish around 200 epochs and the result becomes a completely black image. SPADE does not have this problem, possibly due to the added SPADE residual blocks in the generator. We feel both methods are suitable

Metric	Real Image	pix2pixHD	SPADE
mAP@[.5,.95]	72.4	78.9	79
MAPE	11.6	7.2	9.7
MAE	8.0	4.6	5.5
RMSE	9.6	5.6	6.5

Table 1. Panicle detection and counting results. Each column represents a detection model trained on the corresponding dataset. "Real Image" is the dataset of 400 real images. "pix2pixHD" and "SPADE" are the datasets with 400 real images plus 1000 synthetic images from the corresponding GAN. Bold indicates best performance.

for GAN-based data augmentation for high-resolution UAV images.

5. Conclusion

In this paper, we presented the use of GAN generated synthetic images to augment the training data for panicle detection and counting. We examined two image-to-image translation GANs and showed that their use can improve the performance of panicle detection and counting. We did not use the temporal information available in our real Sorghum UAV dataset during training due to the limitation of the network structures. Future work includes developing multi-temporal methods that can generate synthetic plant images in a temporally consistent style. This will also us to estimate phenotypic traits as the plant grows. We will also examine our approach for estimating traits of other plant such as maize tassels.

6. Acknowledgments

We thank Professor Ayman Habib and the Digital Photogrammetry Research Group (DPRG) from the School of Civil Engineering at Purdue University for providing the images used in this paper. The work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001135. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Address all correspondence to Edward J. Delp, ace@ecn.purdue.edu

References

- [1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. *Proceedings of the International Conference on Machine Learning*, pages 214–223, August 2017. Sydney, Australia. 3
- [2] L. Bi and G. Hu. Improving image-based plant disease classification with generative adversarial network under limited training set. *Frontiers in Plant Science*, 11, 2020. 2, 3

- [3] T. M. Khoshgoftaar C. Shorten. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, June 2019. 2, 3
- [4] E. Cai, S. Baireddy, C. Yang, E. J. Delp, and M. Crawford. Panicle counting in uav images for estimating flowering time in sorghum. *Proceedings of IEEE International Symposium on Geoscience and Remote Sensing*, pages 6280–6283, 2021. Brussels, Belgium. 2
- [5] S. C. Chapman, T. Merz, A. Chan, P. Jackway, S. Hrabar, M. F. Dreccer, E. Holland, B. Zheng, T. J. Ling, and J. Jimenez-Berni. Pheno-copter: A low-altitude, autonomous remote-sensing robotic helicopter for high-throughput field-based phenotyping. *Agronomy*, 4(2):279–301, June 2014. 2
- [6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. . 3, 5
- [7] I. E. Ezeaku and S. G. Mohammed. Character association and path analysis in grain sorghum. *African Journal of Biotechnology*, 5:1337–1340, August 2006. 1, 2
- [8] J. Fargione, J. Hill, D. Tilman, S. Polasky, and P. Hawthorne. Land clearing and the biofuel carbon debt. *Science*, 319(5867):1235–1238, 2008. 1
- [9] R. T. Furbank and M. Tester. Phenomics-technologies to relieve the phenotyping bottleneck. *Trends in Plant Science*, 16(12):635–644, November 2011. 2
- [10] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998. 3
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Proceedings of the Advances in Neural Information Processing Systems*, pages 2672–2680, December 2014. Montreal, Canada. 2, 3
- [12] A. Habib, W. Xiong, F. He, H. L. Yang, and M. Crawford. Improving orthorectification of UAV-Based pushbroom scanner imagery using derived orthophotos from frame cameras. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 262–276, January 2017. 2
- [13] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, October 2017. Venice, Italy. 2
- [14] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Proceedings of the International Conference on Neural Information Processing Systems*, page 3–10, 1993. Denver, Colorado. 3
- [15] P. Isola, J. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, July 2017. Honolulu, HI. 4
- [16] A. Jabbar, X. Li, and B. Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys.*, 54(8), October 2021. 3
- [17] G. Jocher et al. ultralytics/yolov5: v6.1 - TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022. doi: 10.5281/zenodo.6222936. 8
- [18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4396–4405, 2019. Long Beach, CA. 3
- [19] D. Kelly, A. Vatsa, W. Mayham, L. Ngô, A. Thompson, and T. KazicDerek. An opinion on imaging challenges in phenotyping field crops. *Machine Vision and Applications*, pages 1–14, December 2015. 2
- [20] Z. Khan, V. Rahimi-Eichi, S. Haeefe, T. Garnett, and S.J. Miklavcic. Estimation of vegetation indices for high-throughput phenotyping of wheat using aerial imaging. *Plant Methods*, 14(1):20, March 2018. 2
- [21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of the International Conference for Learning Representations*, abs/1412.6980, April 2015. San Diego, CA. 7
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012. 2
- [23] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, page 740–755, 2014. 8
- [24] Z. Lin and W. Guo. Sorghum panicle detection and counting using unmanned aerial system images and deep learning. *Frontiers in Plant Science*, 11:1346, 2020. 2
- [25] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 700–708, December 2017. Long Beach, CA. 3
- [26] P. Lootens, T. Ruttink, A. Rohde, D. Combes, P. Barre, and I. Roldán-Ruiz. High-throughput phenotyping of lateral expansion and regrowth of spaced lolium perenne plants using on-field image analysis. *Plant Methods*, 12(1):32, June 2016. 2
- [27] S. L. Madsen, M. Dyrmann, R. N. Jørgensen, and H. Karstoft. Generating artificial images of plant seedlings using generative adversarial networks. *Biosystems Engineering*, 187:147–159, 2019. 2, 3
- [28] S. Mathur, A. V. Umakanth, V. A. Tonapi, R. Sharma, and M. K. Sharma. Sweet sorghum as biofuel feedstock: recent advances and available resources. *Biotechnology for Biofuels*, 10, 2017. 1
- [29] S. Milz, T. Rüdiger, and S. Süß. Aerial generation: Towards realistic data augmentation using conditional gans. *Proceedings of the 15th European Conference on Computer Vision*, pages 59–72, 01 2019. 3
- [30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014. 3
- [31] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2332–2341, June 2019. Long Beach, CA. 3, 5

- [32] B. B. Pendleton, G. L. Teetes, and G. C. Peterson. Phenology of sorghum flowering. *Crop Science*, 34(5):1263–1266, September 1994. 1, 2
- [33] J. D. Plessis. *Sorghum production*. Department of Agriculture of Republic of South Africa, 2003. 1
- [34] D. Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. 8
- [35] J. R. Prat. *Image-Based Plant Phenotyping Using Machine Learning*. PhD thesis, Purdue University, West Lafayette, IN, May 2019. 1
- [36] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Proceedings of the International Conference on Learning Representations*, November 2016. San Juan, Puerto Rico. 3, 4
- [37] P. S. Rao, R. S. Prakasham, P. P. Rao, S. Chopra, and S. Jose. Sorghum as a sustainable feedstock for biofuels. *Biomass and Biofuels*, pages 27–47, January 2015. 1
- [38] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *Proceedings of The International Conference on Machine Learning*, 48:1060–1069, June 2016. 3
- [39] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1137–1149, June 2016. 2
- [40] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, October 2015. Munich, Germany. 4
- [41] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. 3
- [42] V. Sandfort, K. Yan, P. Pickhardt, and R. Summers. Data augmentation using generative adversarial networks (cycle-gan) to improve generalizability in ct segmentation tasks. *Scientific Reports*, 9:16884, 11 2019. 3
- [43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations*, May 2015. San Diego, CA. 2
- [44] B. Velmurugan, M. Narra, D. M. Rudakiya, and D. Madamwar. *Sweet sorghum: a potential resource for bioenergy production*, pages 215–242. Academic Press, 2020. 1
- [45] A. Walter, F. Liebisch, and A. Hund. Plant phenotyping: from bean weighing to image analysis. *Plant Methods*, 11(1):1–11, 2015. 2
- [46] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, June 2018. Los Alamitos, CA. 3, 4
- [47] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5122–5130, 2017. Honolulu, HA. 3, 5
- [48] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of IEEE International Conference on Computer Vision*, pages 2242–2251, 2017. 3