A THESIS FOR THE DEGREE OF MASTER OF SCIENCE

Generation of 3D paprika (*Capsicum annuum* L.) leaves with edited traits by using deep generative models

심층 생성 모델을 이용한 다양한 특징을 가진
3D 파프리카 잎의 생성

BY

HAYOUNG CHOI

FEBRUARY, 2020

MAJOR IN HORTICULTURAL SCIENCE AND
BIOTECHNOLOGY
DEPARTMENT OF PLANT SCIENCE
GRADUATE SCHOOL
COLLEGE OF AGRICULTURE AND LIFE SCIENCES
SEOUL NATIONAL UNIVERSITY

Generation of 3D paprika (*Capsicum annuum* L.) leaves with edited traits by using deep generative models

UNDER THE DIRECTION OF DR. JUNG EEK SON
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL OF
SEOUL NATIONAL UNIVERSITY

BY
HAYOUNG CHOI

DEPARTMENT OF PLANT SCIENCE
COLLEGE OF AGRICULTURE AND LIFE SCIENCES
SEOUL NATIONAL UNIVERSITY

FEBRUARY, 2020

APPROVED AS A QUALIFIED THESIS OF HAYOUNG CHOI
FOR THE DEGREE OF MASTER OF SCIENCE
BY THE COMMITTEE MEMBERS

CHAIRMAN:

KI SUN KIM, PH.D

VICE-CHAIRMAN: Son, Jung Eek

JUNG EEK SON, PH.D

MEMBER: C. H Chun

CHANGHOO CHUN, PH.D

Generation of 3D paprika (*Capsicum annuum* L.) leaves with edited traits by using deep generative models

Hayoung Choi

Department of Plant Science, Graduate School of Seoul National University

ABSTRACT

Simulation studies using three-dimensional (3D) plant models have been widely used to study the interaction between plant structures and environments. However, the 3D scanned model is more precise than the rule-based model but there is a limit to generate only a static model. The objective of this study was to generate paprika leaves with various morphological traits by using deep generative models and 3D scanned plant models. Paprika (*Capsicum annuum* L.) leaves at 14, 21, 28, 58 days after transplanting were scanned, preprocessed, and then used to train the deep generative models such as variational autoencoder (VAE), generative adversarial network (GAN), and latent space GAN (L-GAN). The optimal number of latent variables in the model was selected via Jensen-Shannon divergence (JSD). The generated leaves were evaluated with JSD, coverage (COV), and minimum matching distance (MMD) to determine the best model for leaf generation: Chamfer distance (CD) and

Earth mover's distance (EMD) were applied to COV and MMD. The best performances were achieved when latent variables were 8, 16, and 8 for VAE, GAN, and L-GAN, respectively. Among the deep generative models, a modified GAN (L-WGAN-EMD) showed the highest performance with JSD = 0.025, MMD-CD = 26.92, MMD-EMD = 14.79, COV-CD = 0.542, and COV-EMD = 0.529. Paprika leaves with various shapes were generated from random latent variables following a normal distribution, and morphological traits of the leaves could be controlled through linear interpolation and simple arithmetic in latent space. The results of this study can be contributed to the applied studies of 3D plant models, such as estimating canopy light interception and photosynthesis, which require detailed but diverse plant structures for realism.

Additional keywords: deep generative model, generative adversarial network, three-dimensional plant model, three-dimensional scanned model, three-dimensional simulation, variational autoencoder

Student Number: 2018-21065

CONTENTS

Page

ABSTRACT i

CONTENTS iii

LIST OF TABLES iv

LIST OF FIGURES v

INTRODUCTION 1

LITERATURE REVIEW 4

MATERIALS AND METHODS 8

RESULTS 23

DISCUSSION 35

CONCLUSION 42

LITERATURE CITED 43

ABSTRACT IN KOREAN 53

LIST OF TABLES

Page

Table 1. Evaluations of the eight generators (variational autoencoder; VAE, generative adversarial network; GAN, and latent space GAN; L-GAN) selected via Jensen-Shannon divergence (JSD) in the test sets. 28

LIST OF FIGURES

Page

Fig. 1. Workflow from selection of deep generative models to leaf generation. 9

Fig. 2. The scanned models of paprika (*Capsicum annuum* L. 'Scirocco') plants at 14, 21, 28, and 56 days after transplanting (DAT). 12

Fig. 3. The histogram of 247 segmented paprika leaves consisting of 226 to 7,291 points, with an average of 3,367. 14

Fig. 4. The examples of the aligned paprika leaves before and after data preprocess. 15

Fig. 5. **A**rchitectures of autoencoder and deep generative models. 20

Fig. 6. Jensen-Shannon Divergence (JSD) and reconstruction loss of the VAE-CD (A) and VAE-EMD (B) according to the number of latent variables (z). 24

Fig. 7. Jensen-Shannon Divergence (JSD) and generator loss according to the number of latent variables (z) in rGAN (A), L-rGAN-CD (B), L-rGAN-EMD (C), WGAN (D), L-WGAN-CD (E), and L-WGAN-EMD (F). 25

Fig. 8. Reconstruction losses of the AE-CD (A) and AE-EMD (B) according to the number of latent variables (z). 26

Fig. 9. Synthetic paprika leaves generated by VAE-CD (A), VAE-EMD (B), L-rGAN-CD (C), L-WGAN-CD (D), L-rGAN-EMD (E), and L-WGAN-EMD (F). 29

Fig. 10. The ground-truths of paprika leaves (A) and the reconstructed models by AE-CD (B), AE-EMD (C), VAE-CD (D), and VAE-EMD (E). 30

Fig. 11. Interpolations of size (A), inclination (B), and curvature (C) of randomly generated leaf point clouds between left and right-most of each row using latent space representation. 32

Fig. 12. Editing leaf point clouds by simple arithmetic in the latent space. 33

Fig. 13. A paprika plant synthesized by applying the generated leaves to existing plant structures. 34

INTRODUCTION

The growth, development and yield of crops in the same environment are somewhat different, because plants adapt to diverse environments by adjusting not only physiological functions but also their structures (Sultan, 2000; Zhu et al., 2015). Three-dimensional simulation studies have been developed and conducted to see how the environment affects plants through plant structures (Evers et al., 2011). Differences in light distribution and light interception as dependent on morphological plant traits and arrangement were examined (Kahlen et al., 2007; Burgess et al., 2015; Tang et al., 2019). Several lighting strategies in combination with a 3D model of tomato were evaluated (De Visser et al., 2014). The impact of plant architecture and canopy connectedness on the movement of predators was examined using virtual plants (Skirvin et al., 2004). As three-dimensional simulation studies have become more active and sophisticated, the necessity of detailed three-dimensional plant models has emerged.

The 3D measurements of plant structures were possible with the use of high-performance computers and the availability of portable cameras and sensors (Paulus, 2019). Unlike rule-based approaches that generate model plants based on user-defined rules from knowledge of plant architectures (Lindenmayer, 1968; Boudon et al., 2012), reconstructed plant models from 3D measurements were constructed sophisticatedly enough to extract the morphological information of the plants and individual organs like leaves or stems (Paulus et al., 2014b; Golbach et al., 2016; Zhang et al., 2016). Thus, simulations could be available with more detailed 3D shapes of plants by digitizing existing structures (Burgess et al., 2015; Retkute et al., 2018, Townsend et al., 2018).

However, there were still challenges for the 3D digitized plant models to be widely used. Sensors and algorithms must overcome the problems of occlusion, plant movement by wind, and the combination of different sensors together (Gibbs et al., 2017; Paulus, 2019). As the plants grow and develop, their size and complexity increase, which leads to time-consuming task. With those reasons, large-scale plant reconstructions and simulations are difficult due to the need to reconstruct crowded scenes containing multiple plants and many leaves (White et al., 2012). Thus, in general, single-plant or few-plant reconstructions were duplicated and randomly rotated for the simulations (Burgess et al., 2015; Retkute et al., 2018; Townsend et al., 2018; Wen et al., 2019), which reduced diversity and realism. Besides, the 3D digitized models are a fixed model at a moment in a particular situation and are difficult to be used for various purposes because they are difficult to transform into other forms.

The deep generative model could be a solution to bring realism and diversity to simulations. The generative model learns data distribution to create new data points from fewer variables, and recently it has been a huge success in just a few years using deep learning (Sajjadi et al., 2017; Cheng et al., 2019). Variational autoencoders (VAEs) (Kingma and Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014) are the most popular and basic deep generative models. The GANs trained in the latent space (L-GAN) are easier to train than general GAN and created 3D objects well (Achlioptas et al., 2017). The objective of this study was to produce paprika leaves with various morphological traits by using deep generative models and the 3D scanned plant models.

LITERATURE REVIEW

3D plant phenotyping

Measuring plants in 3D way has been introduced during the last three decades (Walklate, 1989; Omasa et al., 2007; Kjaer and Ottosen, 2015). 3D plant phenotyping allows researchers to gather morphological information of plants, which enables tracking the geometrical development of the plant (Paproki et al., 2012) and parameterizing of plant organs (Dornbusch et al., 2007). It is crucial to understand the biological and physical processes of the plant growth, which is an important factor in increasing crop yield (Wang et al., 2009). 3D measuring is nondestructive, allowing plants to be monitored over time (Paulus et al., 2014a). In addition, when new features are required later, they can be extracted from the same 3D models, which may not be possible in the 2D approach. In the 2D approach, the features required may not be visible, or the calibration information needed to make real-world measurements might not have been recorded. Therefore, 3D reconstructed models of plants can be used to continuously analyze a wider variety of phenotypes.

Various 3D measurement methods have been used for plant phenotyping (Paulus, 2019). Laser triangulation (LT) uses laser distance measurement and a sensor movement. It has been primarily applied in laboratory environment due to its high resolution and high accuracy measurements (Dupuis and Kuhlmann, 2014), but is currently expensive, time-consuming, and complex to implement

(Zhang et al., 2016). Structure from motion (SfM) uses a set of 2D images to reconstruct a 3D model. It needs a short time for acquiring the 2D images, but its resolution strongly depends on the number of images, the amount of different viewing angles, and camera resolution (Rose et al., 2015). Other techniques, including binocular stereovision (Klodt et al., 2015) and time-of-flight (TOF) (Kloss et al., 2011) are used at various scales to meet the different requirements for plant phenotyping.

The current challenges of 3D plant phenotyping are to scale up from single plant to field scale, to overcome the limitations of the problems of plant movement and occlusion, and to focus the definition of the 3D traits regarding the way traits are measured to enable a comparison of algorithms, plants and treatments among different research groups (Paulus, 2019).

Deep learning in agriculture

Deep learning is a class of machine learning methods based on artificial neural networks. Machine learning algorithms build a mathematical model based on given data, in order to solve problems without being explicitly programmed (Samuel, 1959). Deep learning uses multiple layers to extract features from the raw input with multiple levels of abstraction (LeCun et al., 2015).

Deep learning has recently entered the domain of agriculture for image processing and data analysis. Especially, convolutional neural networks (ConvNet) appear in numerous studies for image process. Plant organs were identified and localized in the 2D images by ConvNet (Pound et al., 2017). Several crop types were also classified, and plant diseases could be detected (Kussul et al., 2017; Mohanty et al., 2016). Besides, environmental factors in greenhouse were interpolated by multilayer perceptron (Moon et al., 2019). it could complete the big data obtained from the greenhouse for further analysis. In addition, some recurrent neural network algorithms have been used for crop classification (Rußwurm and Körner, 2017) and mapping winter vegetation quality (Minh et al., 2018), but deep generative models were rarely used.

Deep generative model

Generative modelling, one of the machine learning areas, learns data distribution to generate new data points with some variations. Training generative models has been a problem for a long time. Classically, most models require strong assumptions about the structure of data, make severe approximations, or rely on computationally expensive inference procedures (Doersch, 2016). More recently, they have achieved great success in just few years by utilizing deep learning. Two of the most popular and efficient approaches are variational autoencoders (VAEs) (Kingma and Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014). Deep generative models have been commonly used for images (Sajjadi et al., 2017; Dolhansky and Canton Ferrer, 2018) and studied for use in audio (Engel et al., 2019) and 3D objects (Achlioptas et al., 2017; Cheng et al., 2019)

MATERIALS AND METHODS

Workflow

Fig. 1 shows the workflow of this study from model selection to the generation of various leaves in this study. The paprika leaves were constructed directly from 3D scanned data. After preprocessing and augmentation, paprika leaves were divided into training set, validation set, and test set. The deep generative models were trained on the training set. The optimal number of latent variables for the models were selected by evaluation of the validation set. All the optimized models were evaluated using the test set and the best deep generative model was determined. The best models not only randomly created leaves but also created leaves with desired traits by manipulating the latent space. Linear interpolation of latent variables led to gradual changes in leaf shape and arithmetic operations in the latent space added or subtracted leaf traits to existing leaves. Diversely generated leaves, along with existing stem information, were used to synthesize whole plant model that could be used for various environmental simulations.

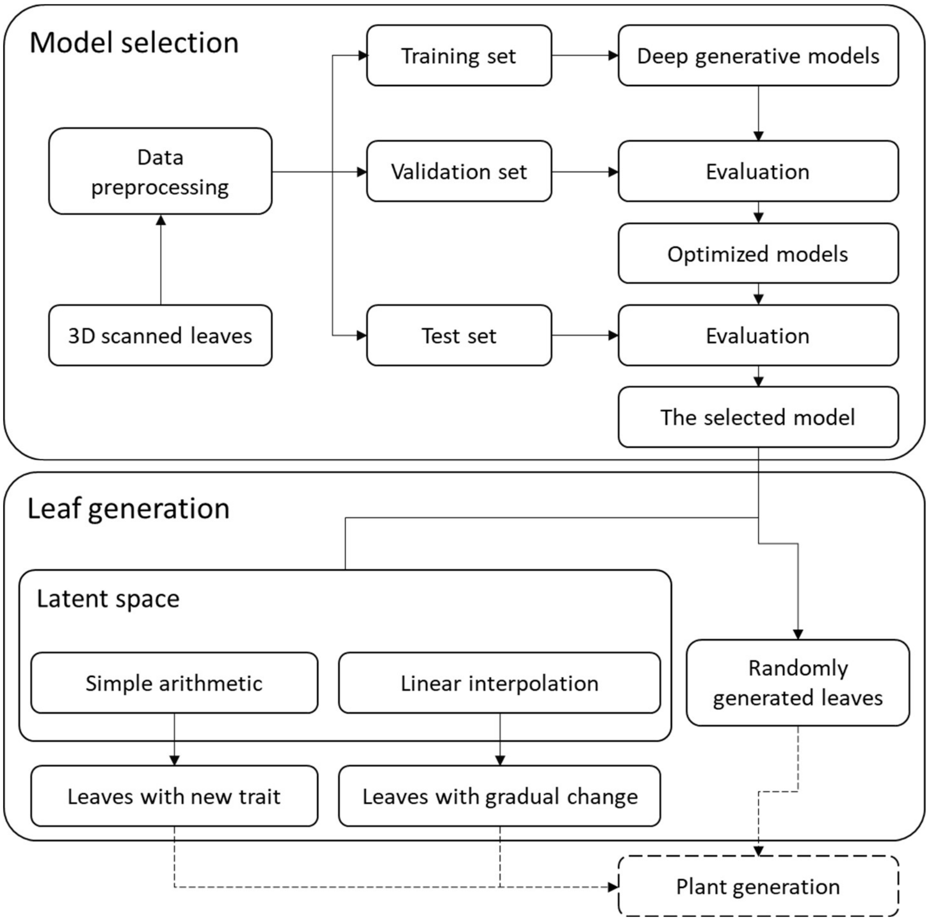

Fig. 1. Workflow from selection of deep generative models to generation of leaves.

Plant material and cultivation conditions

Paprika (*Capsicum annuum* L. 'Scirocco') plants were cultivated in Venlo-type greenhouses of the Protected Horticulture Research Institute, National Institute of Horticultural and Herbal Sciences (RDA), Haman, Korea (35.2ºN, 128.4ºE). The plants were sown on a tray on February 8, 2018, transferred on cubes on March 5, 2018, and finally transplanted on the rock-wool slabs with a planting density of two plants·m^{-2} on April 6, 2018. The plants were pruned to maintain two main stems, which were vertically trellised to a "V" canopy system (Jovicich et al., 2004). Electrical conductivity (EC) of the modified Yamazaki nutrient solutions was 0.8 dS·m^{-1}, gradually increased by 0.2 dS·m^{-1} every week, and finally maintained at 2.5 dS·m^{-1} at the seedling stage. After the transplanting, the nutrient solutions with EC 2.5 dS·m^{-1} and pH 6.0 were supplied 14 times a day at 33 mL per plant by drip irrigation.

3D Reconstruction of paprika leaves

The plants were scanned to reconstruct 3D plant models using a high-resolution portable 3D-scanner (GO!SCAN50^{TM}, CREAFORM, Lévis, Quebec, Canada) at 14, 21, 28, and 56 days after transplanting (DAT) (Fig. 1). A total of eight plants were scanned, two for each DAT. To improve the accuracy of the scan, circular targets with a diameter of 10 mm were attached to the surface of plants. The resolution of scanner was set to 2 mm. After scanning, the point clouds with three-dimensional Cartesian coordinate and RGB color were obtained. Using a 3D reverse engineering software (Geomagic Design X, 3D Systems, Rock Hill, SC, USA), data imperfections like holes, outliers, and overlaps of 3D plant models were corrected and 3D plant models were segmented into 247 leaves consisting of 226 to 7,291 points (Fig. 2).

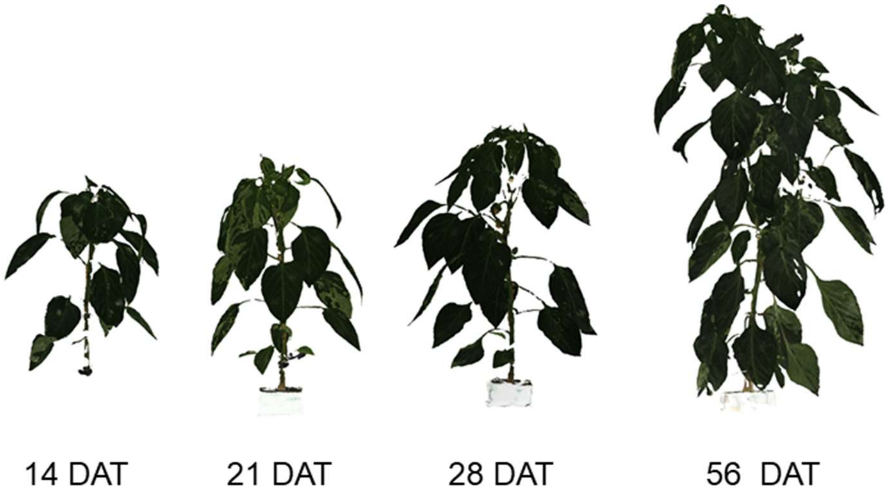

Fig. 2. The scanned models of paprika (*Capsicum annuum* L. 'Scirocco') plants at 14, 21, 28, and 56 days after transplanting (DAT).

Data preprocessing

Observing the segmented leaf point clouds (Fig. 2) and referring to the literature on the generation of three-dimensional objects (Achlioptas et al., 2017), it was decided to keep 2,048 points per leaf to maintain as much information as possible while using as many leaves as possible. Therefore, 2,025 leaf point clouds with 2,048 points per leaf were randomly extracted from segmented leaves, except 22 leaves with fewer than 2,048 points (Fig. 3). They were used for training (70%) and validation (30%) of the generative models. For evaluation of the generated models, 225 different leaves randomly extracted to 2,048 points were used. All leaves were aligned on the z axis through Meshlab (Cignoni et al., 2008). Only x, y, and z coordinates were used as features for model training, and the center coordinates of all the processed leaves were translated to the origin. Min-max normalization (Eq. 1) was used to scale the data between 0 and 1 for making training faster and reducing the chances of getting stuck in local optima.

$$x_{nomalized} = x - x_{min}/x_{max} - x_{min} \tag{1}$$

where, x is the raw data, x_{min} is the minimum values of x, and x_{max} is the maximum values of x.

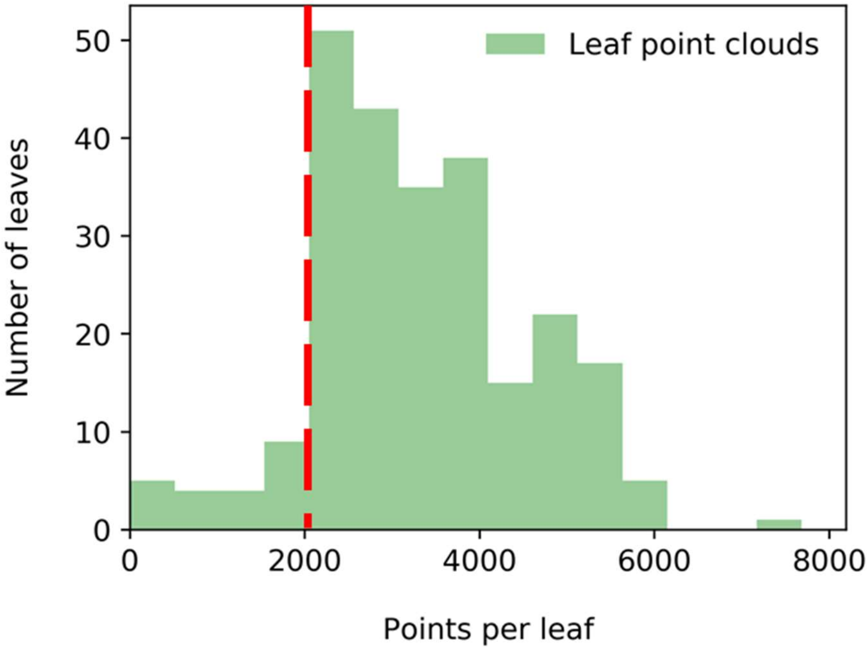

Fig. 3. The histogram of 247 segmented paprika leaves consisting of 226 to 7,291 points, with an average of 3,367. The red dashed line denotes 2,048 points per leaf.

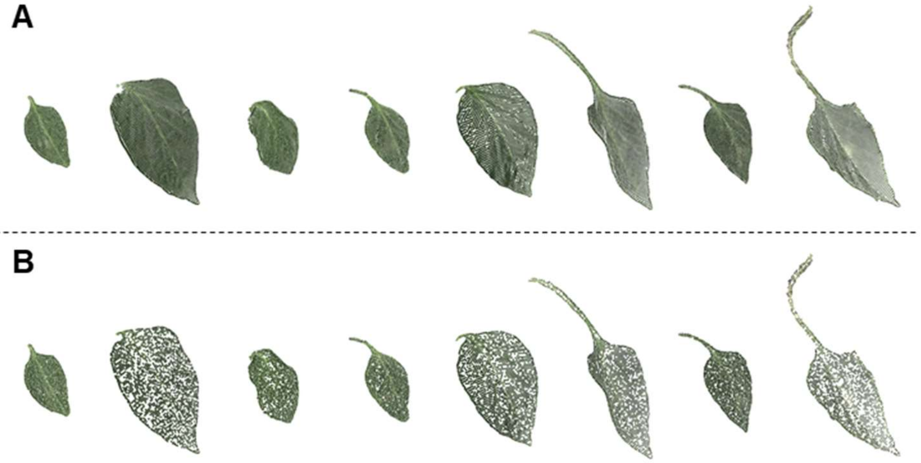

Fig. 4. The examples of the aligned paprika leaves before and after data preprocess. 3D reconstructed leaf models (A) were downsized to consist of 2,048 points (B) for use in the deep generative models.

Deep generative models for leaf generation

Variational autoencoder (VAE), generative adversarial network (GAN), and latent space GAN (L-GAN) were used as generative models. Autoencoder (AE), needed to encode input data for L-GAN, was also trained. VAE consists of two neural networks, an encoder ($q_\theta(z|x)$) and a decoder ($p_\varphi(x|z)$). The encoder takes raw data (x) as input and encode it as a distribution over latent space (z), which size is much smaller than size of raw data. The decoder reconstructs input data (x) given a point sampled from its latent space (z). VAE training uses gradient descent to minimize the loss function ($Loss(\theta,\varphi)$; Eq. 2) with respect to the parameters of the weights of encoder (θ) and decoder (φ).

$$Loss(\theta,\varphi) = -E_{z\sim q_\theta(z|x_i)}\left[\log p_\varphi(x_i|z)\right] + D_{KL}(q_\theta(z|x_i)||p(z)) \qquad (2)$$

The first term is the reconstruction loss that enforces the decoder to learn to reconstruct the data. the second term is a regularizer that prevents overfitting and makes distribution close to a standard normal distribution ($p(z) = Normal(0,1)$). Kulback-Leibler divergence (D_{KL}) is a method of calculating distance between two probability distributions (Kullback and Leibler, 1951).

AE consists of two neural networks, an encoder and decoder like VAE. However, AE has no regularizer in the loss function and does not encode input data as a distribution because it is an architecture that only learns to reproduce its input. The encoder compresses raw data into its latent variables and then, the decoder produces a reconstruction of raw data, from its latent variables.

GAN consists of two neural networks, generator (G) and discriminator (D). GAN samples noise (z) using normal or uniform distribution and uses a generator to create data ($G(z)$). The discriminator distinguishes whether the input of discriminator is real (x) or generated ($G(z)$). The generator tries to minimize the objective function ($V(D,\ G)$; Eq. 3), while the discriminator tries to maximize it.

$$\min_{G}\max_{D} V(D,\ G) = E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z}[\log(1 - D(G(z)))] \quad (3)$$

The formula derives from the cross-entropy between the real and generated distributions. $D(x)$ and $D(G(z))$ are the discriminator's estimates of the probability where real data instance x is real or generated instance $G(z)$ is real.

L-GAN has the same basic structure and training method as GAN but uses latent variables from a pre-trained AE as input data instead of raw data. The generated results of L-GAN are needed to convert by using the AE’s decoder because L-GAN generates latent variables of the AE.

Architectures of deep generative models

In this study, the encoder architecture of AE and VAE followed the design principle of Qi et al. (2017) (Figs. 4A, 4B). Every point was encoded through five 1-D convolutional layers with kernel size 1. Each convolutional layer was followed by a ReLU (Nair and Hinton, 2010) and a batch-normalization layer (Ioffe and Szegedy, 2015). A feature-wise maximum was placed after the convolutions to produce latent space (z-dimensional vector). The decoder transformed the latent variables using four fully-connected layers to produce point clouds. The first of three fully-connected layers had ReLUs and batch-normalization layer.

The CD and the EMD approximation were used as our reconstruction losses, which yielded two distinct AE models (AE-CD and AE-EMD), and two distinct VAE models (VAE-CD and VAE-EMD). For two equally sized point clouds $S_1, S_2 \subseteq R^3$, Chamfer distance (CD; Eq. 4) and earth mover's distance (EMD; Eq. 5, Rubner et al., 2000) are defined by

$$d_{CD}(S_1, S_2) = \sum_{x\in S_1} \min_{y\in S_2} ||x-y||_2^2 + \sum_{y\in S_2} \min_{x\in S_1} ||x-y||_2^2 \qquad (4)$$

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x\in S_1} ||x-\phi(x)||_2 \qquad (5)$$

where $\phi: S_1 \rightarrow S_2$ is a bijection.

The discriminator architecture of GAN (Fig. 4C) had four 1-D convolutional layers with one kernel size and three fully-connected layers without any batch-normalization and with leaky ReLUs (Maas et al., 2013) instead or ReLUs. The output of the last fully-connected layer was fed into a sigmoid function. The generator took as input a Gaussian noise vector and mapped it to a point cloud via four fully-connected layers. The first of three fully-connected layers had leaky ReLUs or ReLUs. In the L-GAN, both the generator and the discriminator were operated on the bottleneck variables of the AE. Thus, no complicated architecture was necessary for L-GAN (Fig. 4D). A generator of a two fully-connected layer coupled with a discriminator of three fully-connected layers sufficed to produce measurably good and realistic results according to literature (Achlioptas et al., 2017). In addition to the general GAN objectives (rGAN, L-rGAN), a WGAN objective (Arjovsky et al., 2017) with gradient penalty (Gulrajani et al., 2017) was used to train GAN and L-GAN (WGAN, L-WGAN).

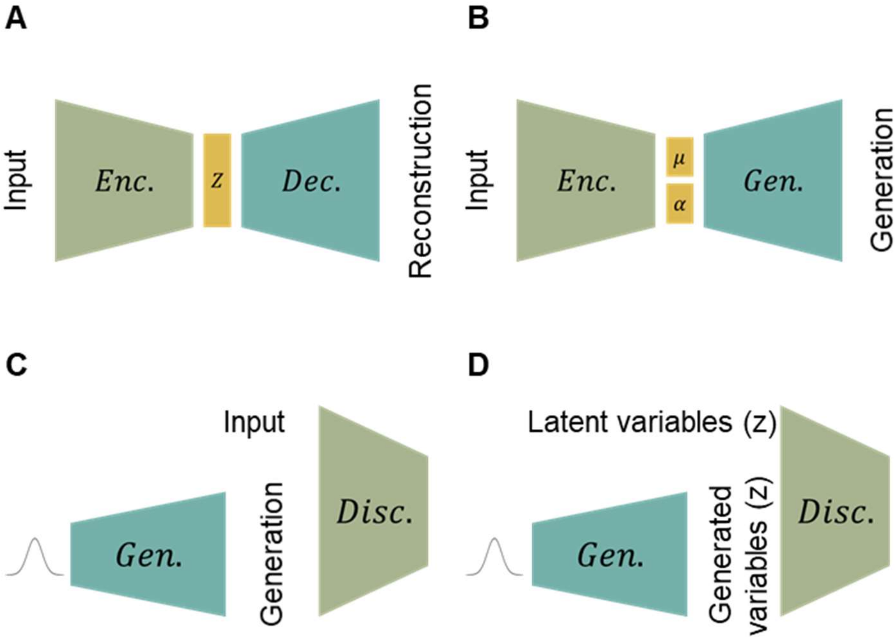

Fig. 5. Architectures of autoencoder and deep generative models. The autoencoder (AE) consists of an encoder (Enc.) and a decoder (Dec.) to compress and reproduce the leaves (A). The Decoder acts as a generator (Gen.) in the variational autoencoder (VAE), which inherits the structures of AE (B). The generative adversarial network (GAN) consists of a generator (Gen.) and a discriminator (Disc.) to generate leaves from random noise (C). Both the generator and the discriminator of the latent space GAN (L-GAN) operate on the latent variables of the AE (D).

Determination of the number of latent variables

To determine an optimal size for the latent space of deep generative models and autoencoder, five architectures of AE, VAE, GAN, and L-GAN with latent vector sizes $k \in \{8, 16, 32, 64, 128\}$, otherwise architecturally identical, were trained. The reconstruction loss was used to select the optimal number of latent variables of AE. In VAE, the number of latent variables can be determined by reconstruction loss and evaluation metrics. However, it is difficult to select models through loss comparison in GAN based model. This is because losses of GAN base model were not directly defined as specific functions, but two neural network structures were competitively trained to produce leaves. Therefore, the evaluations of the generated samples were used to select the optimal number of latent variables of VAE, GAN and L-GAN. To train the deep generative models, the AdamOptimizer (Kingma and Ba, 2014) was used and set to commonly used values. These generative models were trained for a maximum of 1000 epochs. TensorFlow (v. 1.13.1; Abadi et al., 2016) was used for the experiments.

Evaluation metrics

For the evaluation of deep generative models, there has been no consensus on which measure should be used for fair model comparison (Borji, 2019). Therefore, three methods applied to the three-dimensional object from the literature (Achlioptas et al., 2017) were decided to be used: Jensen-Shannon divergence (JSD), Coverage (COV), and Minimum Matching Distance (MMD). JSD evaluates similarity between generated point clouds and reference point clouds. COV is a generative capability in terms of abundance of generated samples, which were measured as fraction of the matched reference point clouds to generated point clouds. Since COV only took the closest point clouds into account, MMD, the minimum distance of every reference point clouds to generated point clouds, was introduced. COV and MMD can be computed using either the CD (COV-CD, MMD-CD) or EMD (COV-EMD, MMD-EMD), respectively. The set of generated point clouds captured all modes of reference point clouds with good fidelity when MMD was small and coverage was large. JSD was correlated well with the MMD in the literature (Achlioptas et al., 2017), so the model was selected based on JSD as the priority in this study. Each generator produced a set of synthetic samples that was equal to the test set, and then, the evaluation was conducted between synthetic samples and test set.

RESULTS

Optimal number of latent variables in deep generative models

The deep generative models with the lowest JSD between the generated leaves and the validation set were selected, as training proceeded. VAE-CD and VAE-EMD, each with eight latent variables, showed the lowest JSD on the training set among all latent variables (Fig. 5). For GAN and L-GAN, 8, 16, 32, 64, 8, and 8 latent variables were fixed for GAN, WGAN, L-rGAN-CD, L-WGAN-CD, L-rGAN-EMD, and L-WGAN-EMD, respectively (Fig. 6, refer to Table 1 for each generative model). The loss and JSD of deep generative models tended to be somewhat similar, but they were not exactly the same and the number of latent variables with minimum values was different.

AE-CD and AE-EMD were also trained to get latent variables to be used as inputs to L-GAN (Fig. 7). The loss of the validation data not used for training was greater than or equal to the loss of the training data. Since this could be a result of overfitting to the training data, the model was selected using the reconstruction loss on the validation set during model training. Thus, 64 and 32 latent variables were fixed on the structures of AE-CD and AE-EMD, respectively.

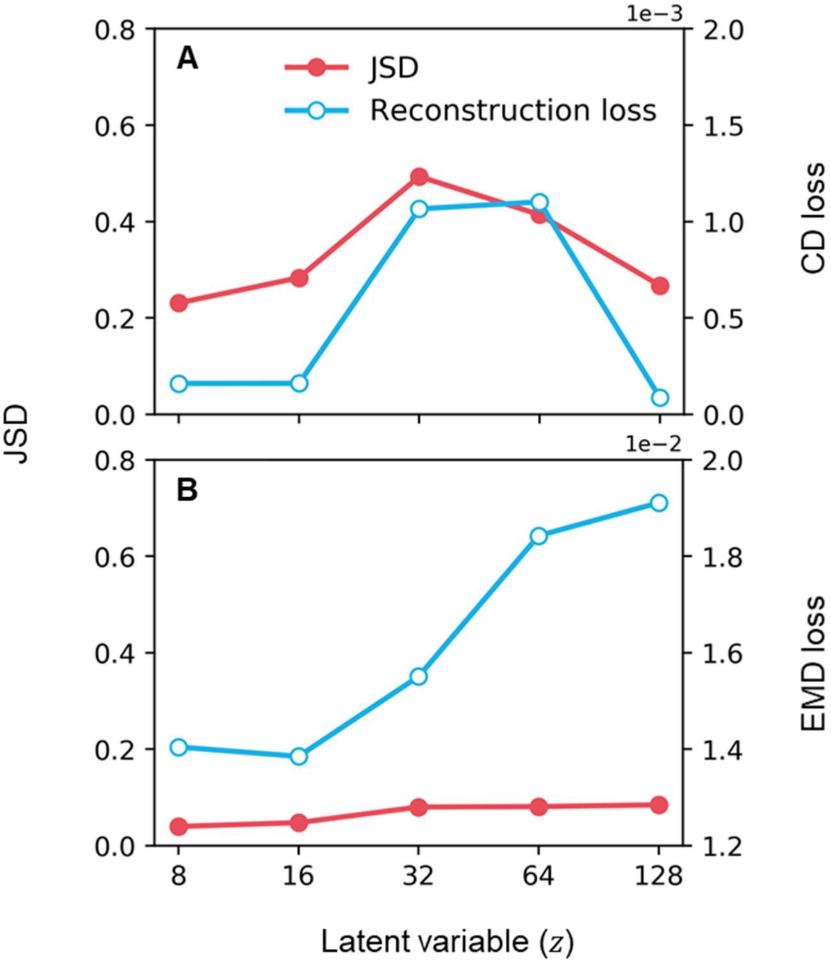

Fig. 6. Jensen-Shannon divergence (JSD) and reconstruction loss of the VAE-CD (A) and VAE-EMD (B) according to the number of latent variables (z). Refer to Table 1 for each generative model.

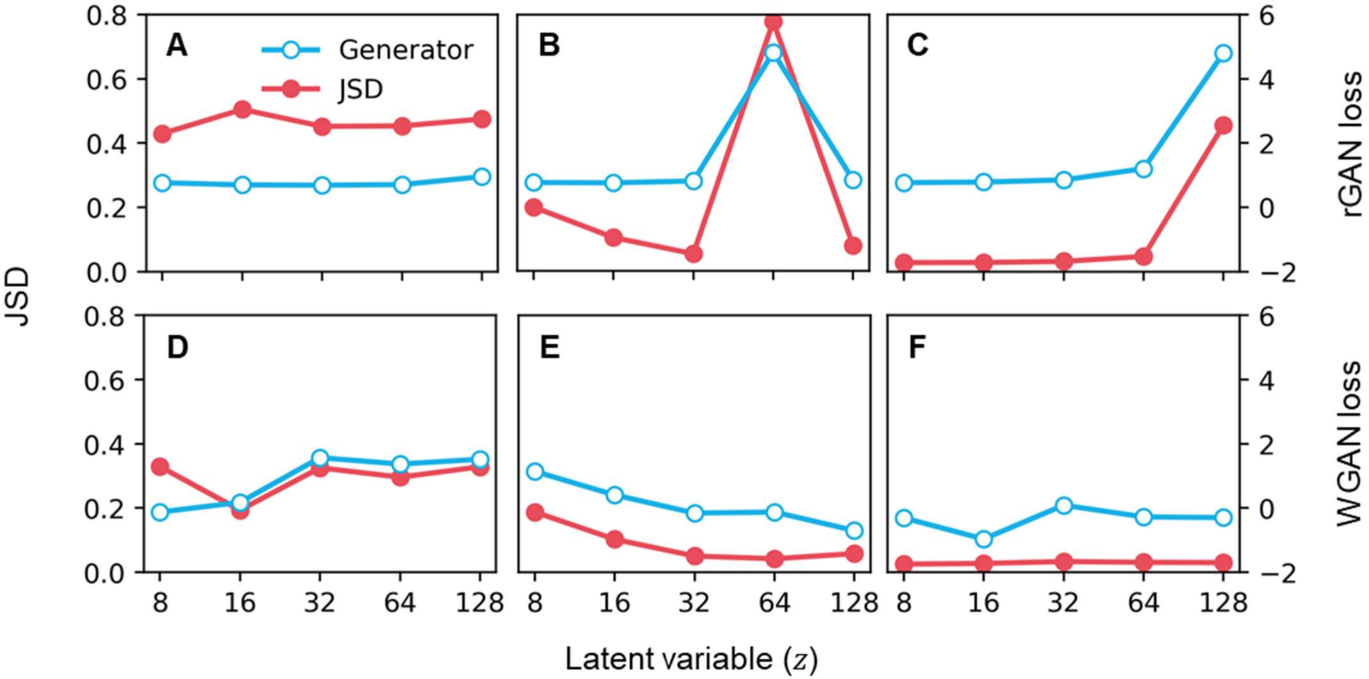

Fig. 7. Jensen-Shannon divergence (JSD) and generator loss according to the number of latent variables (z) in rGAN (A), L-rGAN-CD (B), L-rGAN-EMD (C), WGAN (D), L-WGAN-CD (E), and L-WGAN-EMD (F). Refer to Table 1 for each generative model.

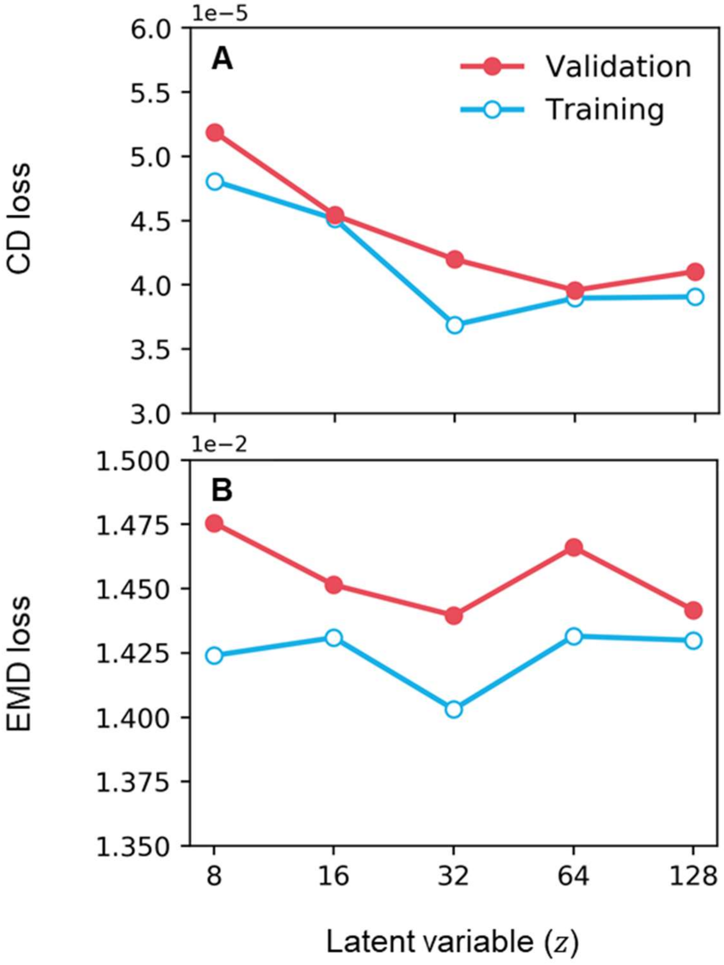

Fig. 8. Reconstruction losses of the AE-CD (A) and AE-EMD (B) according to the number of latent variables (z). Refer to Table 1 for each generative model.

Evaluation of the deep generative models

In Table 1, all types of generators were evaluated based on JSD, MMD, and COV using the test data, after the deep generative models were selected according to the number of latent variables. L-WGAN-EMD performed the best among all evaluation methods. In terms of performance by generator type, L-GAN received the highest score, followed by VAE and then GAN. In most cases, performances of the generative models were higher with EMD loss as a reconstruction loss than CD loss, and with WGAN objective than general GAN objective.

L-GAN showed high quality results in random generation, regardless of the GAN objective or loss of AE (Figs. 8C, 8D, 8E, 8F), while the outputs of rGAN and WGAN were bad (not seen). VAE-CD and VAE-EMD showed poorer performance (not properly created or blurred) for random generation (Figs. 8A, 8B) but showed good reconstruction performance (Figs. 9C, 9D). In addition to the deep generative models, the reconstruction capabilities of the AEs used in the L-GAN were also confirmed. The trained AE-CD and AE-EMD showed high performance with respect to the reconstruction from the data regardless that they were not used for training (Figs. 9A, 9B). All leaves were displayed from the same point of view.

Table 1. Evaluations of the eight generators (variational autoencoder; VAE, generative adversarial network; GAN, and latent space GAN; L-GAN) selected via Jensen-Shannon divergence (JSD) in the test sets.

Generator^{z}	JSD	MMD-CD	MMD-EMD	COV-CD	COV-EMD
VAE-CD	0.234	31.186	27.993	0.520	0.244
VAE-EMD	0.041	30.616	17.314	0.449	0.449
rGAN	0.441	69.611	33.223	0.080	0.027
WGAN	0.205	49.197	29.976	0.178	0.102
L-rGAN-CD	0.050	31.969	18.688	0.471	0.333
L-WGAN-CD	0.046	31.458	17.452	0.400	0.329
L-rGAN-EMD	0.031	28.323	15.130	0.529	0.493
L-WGAN-EMD	0.025	26.915	14.785	0.542	0.529

[z] Two losses of Chamfer distance (CD) and earth mover's distance (EMD) were used for VAE (VAE-CD, and VAE-EMD). General GAN objective and Wasserstein GAN objective were used for GAN-based models (rGAN, WGAN, L-rGAN, and L-WGAN). Since L-GAN uses the latent variables of AE, the loss used by AE were described as L-rGAN-CD, L-WGAN-CD, L-rGAN-EMD, and L-WGAN-EMD.

[y] Lower scores of JSD, MMD (minimum matching distance)-CD and MMD-EMD, and higher scores of COV (coverage)-CD and COV-EMD are better.

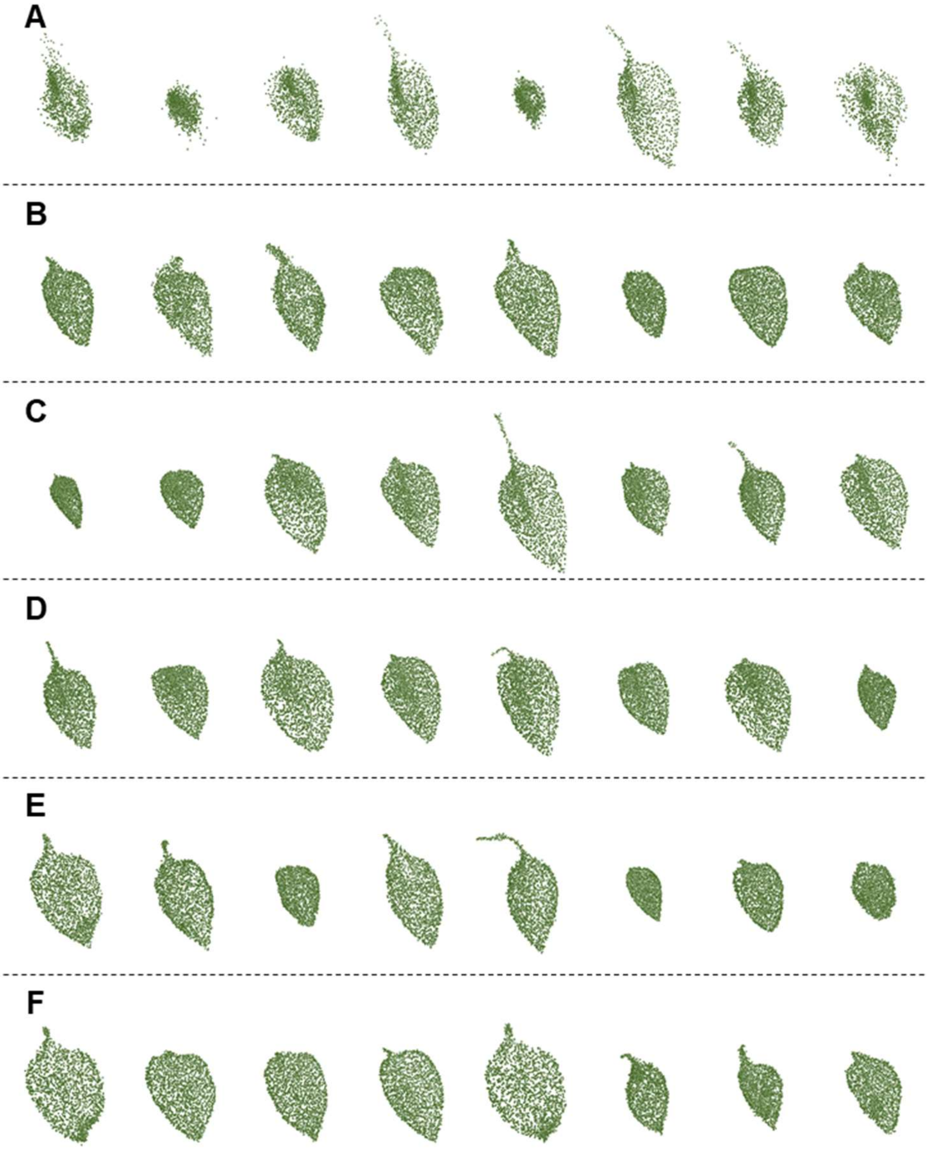

Fig. 9. Synthetic paprika leaves generated by VAE-CD (A), VAE-EMD (B), L-rGAN-CD (C), L-WGAN-CD (D), L-rGAN-EMD (E), and L-WGAN-EMD (F). Refer to Table 1 for each generative model.

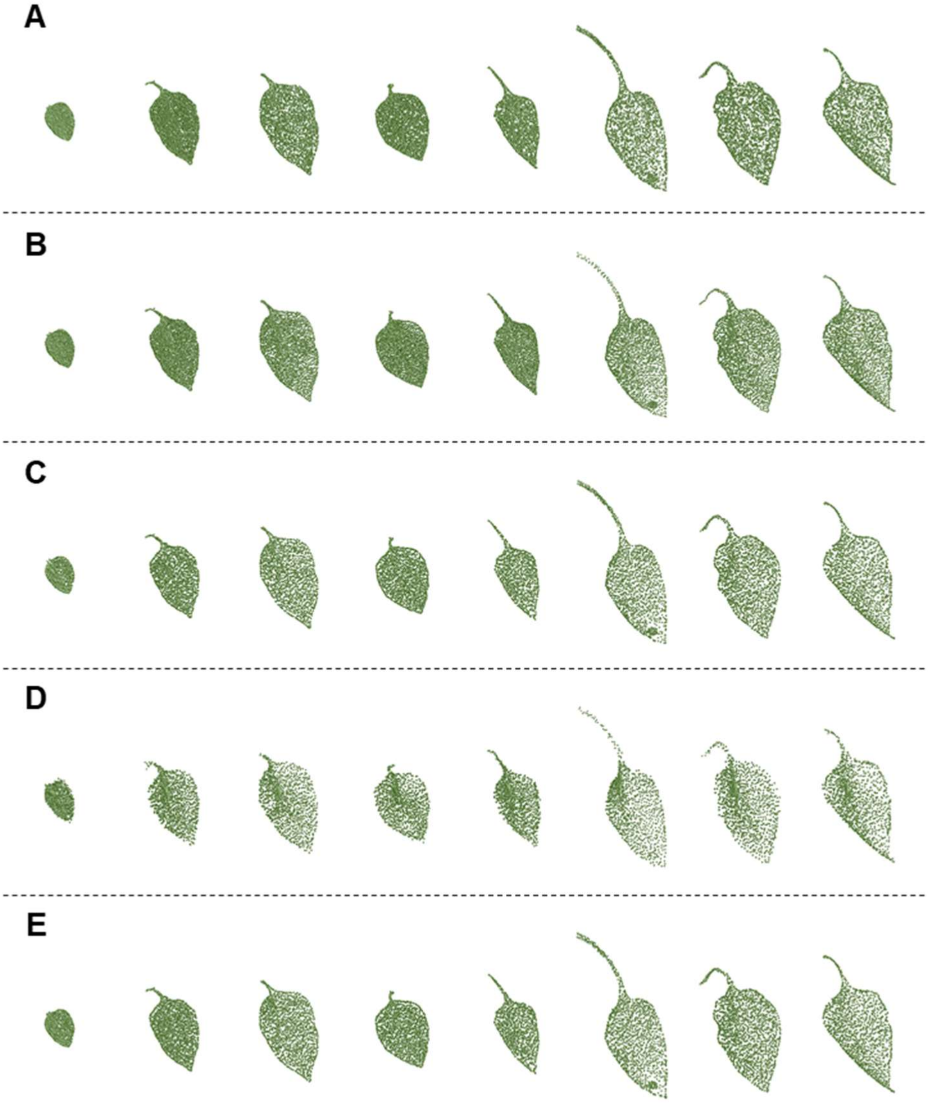

Fig. 10. The ground-truths of paprika leaves (A) and the reconstructed models by AE-CD (B), AE-EMD (C), VAE-CD (D), VAE-EMD (E). Refer to Table 1 for each generative model.

Generation of the leaves and constructions of the whole plant

The well-trained latent representations could generate good samples based on latent variables created by performing interpolation or simple arithmetic. Linear interpolations in the latent space produced smooth transition between different sizes, inclinations, and curvatures of leaves, and showed good results (Fig. 10). The semantic components were obtained from the difference in the latent space between the two leaves, which could be added to other latent variables of leaves to create leaves with new traits (Fig. 11). Leaves were enlarged, inclined, or curved down by simple arithmetic.

Meshes was constructed from leaf point clouds using screened Poisson surface reconstruction (Kazhdan and Hoppe, 2013), and then, leaves with faces were created after few post-processing on the meshes. The whole paprika plant was synthesized by replacing the existing leaves with variously generated leaves in the existing plant stem structure (Fig. 12). Although there were some unnatural aspects of the connection between the leaf and the stem, overall realistic paprika was produced.

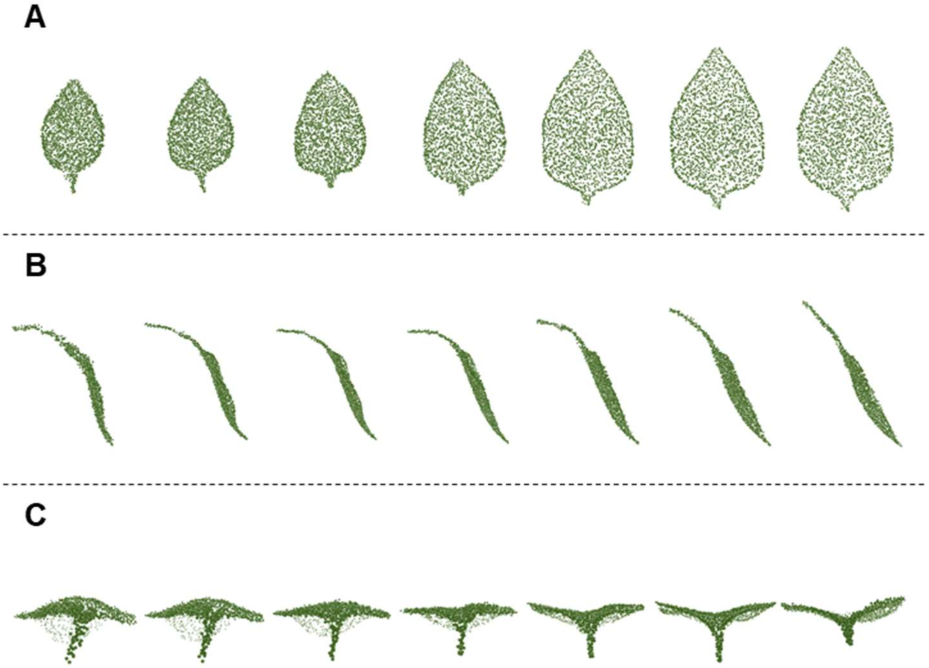

Fig. 11. Interpolations of size (A), inclination (B), and curvature (C) of randomly generated leaf point clouds between left and right-most of each row using latent space representation.

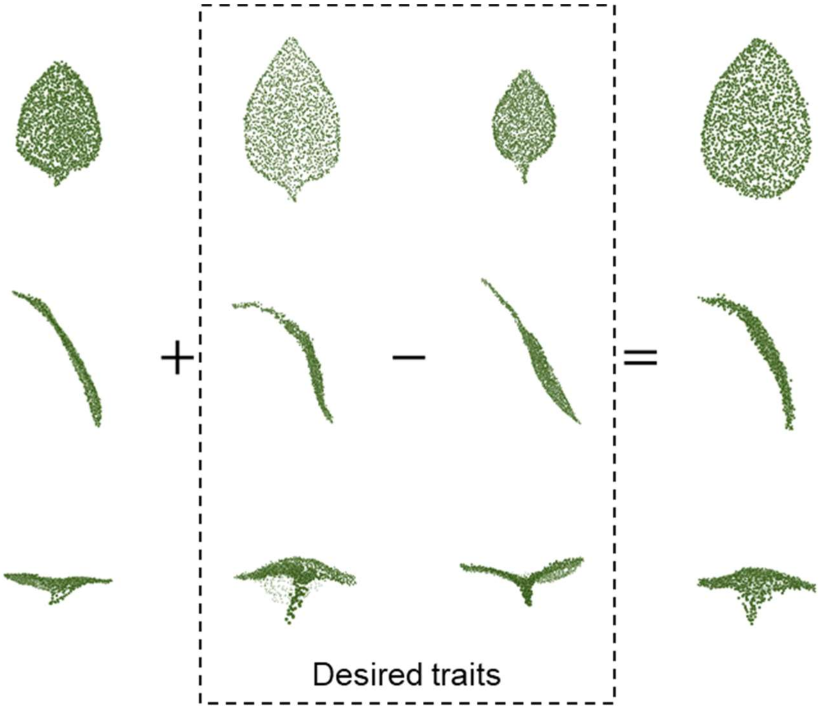

Fig. 12. Editing leaf point clouds by simple arithmetic in the latent space. Various traits could be imparted to generated leaves, such as size, inclination, and curvature.

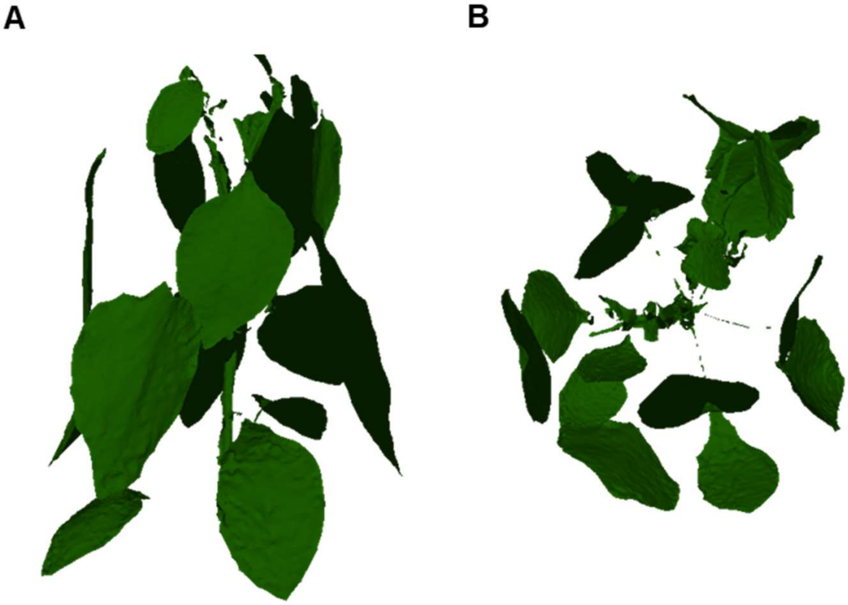

Fig. 13. A paprika plant synthesized by applying the generated leaves to existing plant structures: front view (A) and top view (B).

DISCUSSION

Optimal design of the deep generative models

The optimal number of latent variables for deep generative models were determined through the evaluation (Figs. 5, 6). The well-trained deep generative models, except L-rGAN-CD (32 latent variables) and L-WGAN-CD (64 latent variables), showed the highest performance in very small number of latent variables (8 or 16 latent variables). In general, about 100 latent variables in a GAN-based model were preferred for generating output (Denton et al., 2015; Radford et al., 2015; Achlioptas et al., 2017). This is because the existing models had a reasonable accuracy when using 100 latent variables, and the purpose of studies was to make it easier to compare with other models by fixing the latent variables rather than using the latent variables. In this study, various latent variables were tested for better performance and utilization of latent variables, and the better results were obtained from few variables, which facilitate the adjustment of latent variables. GAN generally cannot extract latent variables from the original, but L-GAN used latent variables of AE that were already compressed and characterized. Thus, the good results were generated from a small number of latent variables of L-GAN.

The optimal VAE was selected the by evaluating the generated leaves, even though a loss function for VAE was defined explicitly. Because, when selected through reconstruction loss, VAE-EMD showed similar results as selected through evaluation, but VAE-CD showed very low results. In the training of VAE-CD, the scale of CD loss differs not only from the scale of regularizer, but also the convergence rate of the reconstructions loss differs from that of the KL divergence used as a regularizer. As a result, regularization of latent space is poor, resulting in high reconstruction performance but relatively low random generation performance when VAE-CD was selected by loss function.

In this study, the optimized structure was selected by adjusting only the number of latent variables, while the rest of the structure was the same. It might be needed to use different hyperparameters of the architectures like the number of neurons and layers for improving the performance of deep generative models. In order to do that, many experiments must be repeated to find the optimal hyperparameters for deep learning models. Finding the optimal model structure and searching for hyperparameter is called automated machine learning (AutoML). Recently, many researches have been conducted to solve this process by deep learning. NAS (Neural Architecture Search; Zoph and Le, 2016) and NASNet (Zoph et al., 2018) searched the neural network structure using reinforcement learning. However, since most AutoMLs have been mainly used for image classification or recurrent cell, they were currently a bit difficult to apply them directly to the generative model. Thus, the model was determined by modifying only the latent variables that make up the latent representations.

Performance of the deep generative models

The L-WGAN-EMD (eight latent variables) highly rated in all evaluation methods could generate good-looking paprika leaves (Fig. 8F). Other L-GANs were slightly lower in rating than L-WGAN (EMD) but showed visually high performance (Figs. 8C, 8D, 8E). L-GANs operate on latent variables of the AE, not directly using raw point cloud, which is directly related to AE performance. If the performance of the AE is good, the faster training speed and higher performance would be achieved because L-GAN learns the extracted feature data of smaller size. In this study, both AE-CD and AE-EMD had high performance (Fig.9), so the L-GANs were able to output good results.

End-to-end GANs are generally difficult to train as this study showed ((Figs. 6A, 6D; Table 1). In general, GAN-based models typically suffer from training disorders like non-convergence, model collapse, and diminished gradient (Mao et al., 2017; Arjovsky et al., 2017). This was well observed when the general GAN (rGAN) objective was used (Figs. 6A, 6B, 6C). To solve this problem, many subsequent studies emerged to find an objective function with smoother and non-vanishing gradients (Mao et al., 2017; Arjovsky et al., 2017; Gulrajani et al., 2017; Berthelot et al., 2017). The WGAN, one of the solutions, showed good results and there was no collapse in model training in this study (Fig. 6, Table 1). There was no absolutely superior function over the others (Qin et al., 2018), so even if other functions were used, overwhelmingly better performances would not be seen.

Unlike good performance in reconstruction, the lower performance of VAE than L-GAN in generating leaves was observed and VAE-CD performed worse than VAE-EMD (Table 1, Fig. 8). In VAE, one noise was drawn from the zero-mean Gaussian and calculated with the mean and variance from encoder to form a sampled latent vector z, which is called a reparameterization trick (Kingma and Welling, 2013). Because of the reparameterization trick, z may be different even if the input data is the same, so the final output was not clear and blurry (Figs. 8A, 8B).

In this study, two distance metrics (CD and EMD) were used as reconstruction losses. CD made it difficult to penalize the point clouds that were over-populated in locations (Achlioptas et al., 2017). If such a trend occurs during model training, the generative model may be poorly trained. In the study, no such phenomenon was observed visually in the generated samples of the selected models, except that the overpopulation of points was found in the result from the VAE-CD (Figs. 8A, 9D). EMD that effectively solved an assignment problem was differentiable almost everywhere and showed good result when used as reconstruction loss (Fan et al., 2017; Achlioptas et al., 2017). In practice, computation of EMD is too expensive for deep learning, even on graphics hardware. Since the computational cost of CD metrics is less than that of EMD metrics, L-GANs-CD may be a good choice for simple plant simulations that do not require high accuracy and seek high efficiency.

Availability of the generated leaf models

Currently, the resolution of scanners is sufficient to present paprika plants, but due to the movement of plants by wind and the nature of laser scan, the edges of plant parts are poorly scanned and require further processing (Klapa and Mitka, 2017). In particular, petioles were not properly scanned so that the leaf data without petiole quite existed in this experimental data (Fig. 3). Despite the increase in diversity of leaves due to the incompleteness of the data, L-GAN-EMD showed high performance with or without petiole (Table 1, Fig. 9). This indicates that the deep generative model can show high performance even when the entire leaf shape is expressed through the development of 3D measurements on the plant.

The manipulation in the latent space allows the model possible to express diverse shape of the leaves, suggesting that it could be used to simulate realistic three-dimensional growth and morphological changes of leaves. Growth estimation of leaves is crucial to analysis of plant growth and yield, transpiration and photosynthesis (Rajcan and Tollenaar, 1999; Rawson and Hindmarsh, 1983). Most plants display strong morphological changes during their diurnal and seasonal development, which depend on the availability of resources and on the fluctuation of abiotic factors (Long et al., 2006). Since the entire plant was generated through replacing the existing leaves with generate leaves in the existing structure, the direction of the leaves and the stem structure were not changed (Fig. 12). Thus, no matter how well the model has trained the distribution of the original leaf data, the distribution expressed at the plant level were limited. At this time, it is expected to resolve these problems by using the rule-based model for the structure and shape of the stem or to apply the generative model to generate various stems. Parametric L-system rules were used to construct branch system for 3D tree models (Tang et al., 2019). Also, more natural plants could be possible through supervised learning that provides the model with information about the shape according to the leaf position. If generative model can be trained to generate whole plant models, it will yield more realistic results. However, it currently seems to be challenging due to technical and data issues.

The trained L-GAN-EMD could create various leaves of paprika randomly as shown by the high coverage in the evaluation (Table 1, Fig. 9), and create some desired leaves by modifying latent variables (Figs. 10, 11). However, several latent variables act on certain trait and each of the latent vector has normally no special meaning, so time-consuming task is required to get the desired leaves. InfoGAN that gives meaning to each latent variable could reduce task for finding the desired result, thus providing convenience for interpretation of latent space (Chen et al., 2016). However, in this study, since the leaves are represented by sufficiently small latent variables, the characteristics of the leaves are not difficult to manipulate.

CONCLUSION

In this study, realistic paprika leaves were created by using deep generative models. Among the deep generative models, the modified GAN (L-WGAN-EMD) had the highest performance. The trained model could generate leaves from eight latent variables that were much smaller than the leaf data size. Arithmetic operations and gradual changes on the latent space could lead to modification of morphological leaf traits. A paprika plant was synthesized by applying the generated leaves to existing plant structures. The results of this study can be contributed to the applied studies of 3D plant models, such as estimating canopy light interception and photosynthesis, which require detailed but diverse plant structures for realism. For more complete representations, deep generative models for randomly generating stem structure and leaves of plants.

LITERATURE CITED

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Achlioptas P, Diamanti O, Mitliagkas I, Guibas L (2017) Learning representations and generative models for 3d point clouds. arXiv preprint arXiv:1707.02392.

Aguilar JJ, Torres F, Lope MA (1996) Stereo vision for 3D measurement: accuracy analysis, calibration and industrial applications. Measurement, 18:193-200.

Arjovsky M, Chintala S, Bottou L (2017) Wasserstein gan. arXiv preprint arXiv:1701.07875.

Burgess AJ, Retkute R, Pound MP, Foulkes J, Preston SP, Jensen OE, Pridmore TP, Murchie EH (2015) High-resolution three-dimensional structural data quantify the impact of photoinhibition on long-term carbon gain in wheat canopies in the field. Plant Physiol 169:1192-1204.

Berthelot D, Schumm T, Metz L (2017) Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717.

Bertheloot J, Cournède PH, Andrieu B (2011) NEMA, a functional–structural model of nitrogen economy within wheat culms after flowering.

I. Model description. Ann Bot 108:1085-1096.

Bertheloot J, Wu Q, Cournède PH, Andrieu B (2011) NEMA, a functional–structural model of nitrogen economy within wheat culms after flowering. II. Evaluation and sensitivity analysis. Ann Bot 108:1097-1109.

Boudon F, Pradal C, Cokelaer T, Prusinkiewicz P, Godin C (2012) L-Py: an L system simulation framework for modeling plant architecture development based on a dynamic language. Front Plant Sci 3:76.

Borji A (2019) Pros and cons of GAN evaluation measures. Comput Vis Image Underst 179:41-65.

Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel, P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Adv Neural Inf Process Syst pp. 2172-2180.

Cheng S, Bronstein M, Zhou Y, Kotsia I, Pantic M, Zafeiriou S (2019) MeshGAN: Non-linear 3D Morphable Models of Faces. arXiv preprint arXiv:1903.10384.

Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G (2008) Meshlab: an open-source mesh processing tool. Eurographics Italian Chapter Conference pp.129-136.

Denton EL, Chintala S, Fergus R. (2015) Deep generative image models using a laplacian pyramid of adversarial networks. Adv Neural Inf Process Syst pp. 1486-1494.

Doersch C (2016) Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.

Dolhansky B, Canton Ferrer C (2018) Eye in-painting with exemplar generative adversarial networks. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit pp.7902-7911.

Dornbusch T, Wernecke P, Diepenbrock W (2007) A method to extract morphological traits of plant organs from 3D point clouds as a database for an architectural plant model. Ecol Model 200:119-129.

Dupuis J, Kuhlmann H (2014) High-precision surface inspection: uncertainty evaluation within an accuracy range of 15 μm with triangulation-based laser line scanners. J Appl Geodesy 8:109-118.

Engel J, Agrawal KK, Chen S, Gulrajani I, Donahue C, Roberts A (2019) Gansynth: Adversarial neural audio synthesis. arXiv preprint arXiv:1902.08710.

Evers JB, van der Krol AR, Vos J, Struik PC (2011) Understanding shoot branching by modelling form and function. Trends Plant Sci 16:464-467.

Fan H, Su H, Guibas LJ (2017) A point set generation network for 3d object reconstruction from a single image. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit pp.605-613.

Gibbs JA, Pound M, French AP, Wells DM, Murchie E, Pridmore T (2017) Approaches to three-dimensional reconstruction of plant shoot topology and geometry. Funct Plant Biol 44:62-75.

Godin C (2000) Representing and encoding plant architecture: a review. Ann

For Sci 57:413-438.

Golbach F, Kootstra G, Damjanovic S, Otten G, van de Zedde R (2016) Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. Mach Vis Appl 27:663-680.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville Aaron, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst pp.2672-2680.

Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of wasserstein gans. Adv Neural Inf Process Syst pp.5767-5777.

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167

Jovicich E, Cantliffe DJ, Stoffella PJ (2004) Fruit yield and quality of greenhouse-grown bell pepper as influenced by density, container, and trellis system. HortTechnology 14:507-513

Kazhdan M, Hoppe H (2013) Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) 32:29.

Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv prepr arXiv:1412.6980v9

Kjaer, K, Ottosen, CO (2015) 3D laser triangulation for plant phenotyping in challenging environments. Sensors 15:13533-13547.

Klapa P, Mitka B (2017) Edge effect and its impact upon the accuracy oF 2d and 3d modelling using laser scanning. Geomat Landmanag Landsc 1:25–33.

Klodt M, Herzog K, Töpfer R, Cremers D (2015) Field phenotyping of grapevine growth using dense stereo reconstruction. BMC Bioinformatics 16:143.

Klose R, Penlington J, Ruckelshausen A (2011) Usability study of 3D time-of-flight cameras for automatic plant phenotyping. Image Analysis for Agricultural Products and Processes 69:93–105

Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22:79-86.

Kussul N, Lavreniuk M, Skakun S, Shelestov A (2017) Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci Remote Sens Lett 14:778–782.

Lindenmayer A (1968) Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. J Theor Biol 18:280–299.

Long SP, ZHU XG, Naidu SL, Ort DR (2006) Can improvement in photosynthesis increase crop yields? Plant Cell Environ 29:315-330.

Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. Proc. icml 30:3.

Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. Proc IEEE Int Conf Comput Vis pp. 2794-2802.

Minh DHT, Ienco D, Gaetano R, Lalande N, Ndikumana E, Osman F, Maurel P (2018) Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1. IEEE Geosci Remote Sens Lett 15:464-468.

Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419

Moon T, Hong S, Choi HY, Jung DH, Chang SH, Son JE (2019) Interpolation of greenhouse environment data using multilayer perceptron. Comput Electron Agr 166:105023.

Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. Proc Int Conf Mach Learn pp.807-814.

Omasa K, Hosoi F, Konishi A (2007) 3D lidar imaging for detecting and understanding plant responses and canopy structure. J Exp Bot 58:881–898.

Paproki A, Sirault X, Berry S, Furbank R, Fripp J (2012) A novel mesh processing based technique for 3D plant analysis. BMC Plant Biol 12:63.

Paulus S (2019) Measuring crops in 3D: using geometry for plant phenotyping. Plant Methods 15:103.

Paulus S, Dupuis J, Riedel S, Kuhlmann H (2014) Automated analysis of barley organs using 3D laser scanning: An approach for high throughput phenotyping. Sensors 14:12670-12686.

Paulus S, Schumann H, Kuhlmann H, Léon J (2014) High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants. Biosyst Eng 121:1-11.

Pound MP, Atkinson JA, Townsend AJ, Wilson MH, Griffiths M, Jackson AS, Bulat A, Tzimiropoulos G, Wells DM, Murchie EH (2017) Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. Gigascience 6:gix083.

Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit pp.652-660.

Qin Y, Mitra N, Wonka P (2018) Do GAN Loss Functions Really Matter?. arXiv preprint arXiv:1811.09567.

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

Rajcan I, Tollenaar M (1999) Source: sink ratio and leaf senescence in maize:: I. Dry matter accumulation and partitioning during grain filling. Field Crops Res 60:245-253.

Rawson, HM, Hindmarsh, JH (1983) Light, leaf expansion and seed yield in sunflower FUNCT PLANT BIOL 10:25-30.

Retkute R, Townsend AJ, Murchie EH, Jensen OE, Preston SP (2018) Three-dimensional plant architecture and sunlit–shaded patterns: A stochastic model of light dynamics in canopies. Ann Bot 122:291-302.

Rose JC, Paulus S, Kuhlmann H (2015) Accuracy analysis of a multi-view stereo approach for phenotyping of tomato plants at the organ level. Sensors 15:9651–9665.

Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis 40:99-121.

Rußwurm M, and Körner M (2017) Multi-temporal land cover classification with long short-term memory neural networks. Int Arch Photogramm, Remote Sens Spatial Inform Sci 42:551.

Sajjadi MS, Scholkopf B, Hirsch M (2017) Enhancenet: Single image super-resolution through automated texture synthesis. Proc IEEE Int Conf Comput Vis pp.4491-4500.

Samuel AL (1959) Some studies in machine learning using the game of checkers. IBM J Res Dev 3:210-229

Sultan SE (2000) Phenotypic plasticity for plant development, function and life history. Trends Plant Sci 5:537-542.

Tang L, Yin D, Chen C (2019) Optimal Design of Plant Canopy Based on Light Interception: A Case Study With Loquat. Front Plant Sci 10:364.

Townsend AJ, Retkute R, Chinnathambi K, Randall JW, Foulkes J,

Carmo-Silva E, Murchie EH (2018) Suboptimal acclimation of photosynthesis to light in wheat canopies. Plant Physiol 176:1233-1246.

Vos J, Evers JB, Buck-Sorlin GH, Andrieu B, Chelle M, De Visser PH (2009) Functional–structural plant modelling: a new versatile tool in crop science. J Exp Bot 61:2101-2115.

Walklate PJ (1989) A laser scanning instrument for measuring crop geometry. Agr Forest Meteorol 46:275-284.

Wang H, Zhang W, Zhou G, Yan G, Clinton N (2009) Image-based 3D corn reconstruction for retrieval of geometrical structural parameters. Int J Remote Sens 30:5505–5513.

Wen W, Guo X, Li B, Wang C, Wang Y, Yu Z, Wu S, Fan J, Gu S, Lu X (2019) Estimating canopy gap fraction and diffuse light interception in 3D maize canopy using hierarchical hemispheres. Agr Forest Meteorol 276: 107594.

Zhang Y, Teng P, Shimizu Y, Hosoi F, Omasa K (2016) Estimating 3D leaf and stem shape of nursery paprika plants by a novel multi-camera photography system. Sensors 16:874.

Zhu J, van der Werf W, Anten NP, Vos J, Evers JB (2015) The contribution of phenotypic plasticity to complementary light capture in plant mixtures. New Phytol 207:1213-1222.

Zoph B, Le QV (2016) Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit pp.8697-8710.

ABSTRACT IN KOREAN

3 차원 식물모델을 활용한 시뮬레이션은 식물 구조와 환경과의 상호 작용을 연구하기 위해서 광범위하게 사용되고 있다. 그러나 3 차원 스캔된 식물 모델은 규칙 기반의 모델에 비하여 정밀하지만, 정적 모델만을 생성할 수 있는 한계가 있었다. 본 연구의 목적은 3 차원 스캔 식물모델과 심층 생성 모델을 활용하여 다양한 형태적 특징을 가진 파프리카 （*Capsicum annuum* L.） 잎을 생성하는 것이다. 정식 후 14, 21, 28, 58 일의 잎을 3 차원 스캐너로 스캔 후 전 처리하여 생성 모델(variational autoencoder, VAE; generative adversarial network, GAN; latent space GAN, L-GAN)의 학습데이터로 사용하였다. 모델의 최적 잠재 변수 수는 Jensen-Shannon divergence (JSD)를 이용하여 결정하였다. 각 모델이 생성한 잎에 대해 JSD 와 Chamfer distance(CD) 및 Earth mover’s distance(EMD)를 적용한 coverage(COV) 및 minimum matching distance(MMD)를 평가하여 최적의 심층 생성 모델을 결정하였다. 잠재변수의 수는 VAE 8 개, GAN 16 개, L-GAN 8 개에서 가장 최적의 성능을 보였다. 여러 심층 생성 모델 중 L-WGAN-EMD 가 가장 높은 성능(JSD=0.025, MMD-CD=26.92, MMD-EMD=14.79, COV-CD=0.542, COV-EMD=0.529)을 보였다. 학습된 심층

생성 모델은 정규 분포를 따르는 무작위 잠재변수로부터 다양한 형태를 가지는 잎을 생성하였고, 잠재 공간 내 선형 보간 및 간단한 산술 연산을 통해서 잎의 형태적 특징을 조절할 수 있었다. 본 연구 결과는 추후 군락의 수광 분포 및 광합성속도 추정 등 정확하면서도 다양한 형태의 식물구조가 필요한 3 차원 식물 모델의 응용 연구에 기여할 것으로 사료된다.

추가 주요어: 3 차원 스캔 모델, 3 차원 시뮬레이션, 3 차원 식물모델, 변분적 오토인코더 (VAE), 생성적 적대 신경망 (GAN), 심층 생성 모델

학 번: 2018-21065