Separating the Structural Components of Maize for Field Phenotyping Using Terrestrial LiDAR Data and Deep Convolutional Neural Networks

Shichao Jin, Yanjun Su, Shang Gao, Fangfang Wu, Qin Ma, Kexin Xu, Qin Ma, Tianyu Hu, Jin Liu, Shuxin Pang, Hongcan Guan, Jing Zhang, and Qinghua Guo

***Abstract*— Separating structural components is important but also challenging for plant phenotyping and precision agriculture. Light detection and ranging (LiDAR) technology can potentially overcome these difficulties by providing high quality data. However, there are difficulties in automatically classifying and segmenting components of interest. Deep learning can extract complex features, but it is mostly used with images. Here, we propose a voxel-based convolutional neural network (VCNN) for maize stem and leaf classification and segmentation. Maize plants at three different growth stages were scanned with a terrestrial LiDAR and the voxelized LiDAR data were used as inputs. A total of 3000 individual plants (22 004 leaves and 3000 stems) were prepared for training through data augmentation, and 103 maize plants were used to evaluate the accuracy of classification and segmentation at both instance and point levels. The VCNN was compared with traditional clustering methods (*K*-means and density-based spatial clustering of applications with noise), a geometry-based segmentation method, and state-of-the-art deep learning methods (PointNet and PointNet++). The results showed that: 1) at the instance level, the mean accuracy of classification and segmentation (F-score) were 1.00 and 0.96, respectively; 2) at the point level, the mean accuracy of classification and segmentation (F-score) were 0.91 and 0.89, respectively; 3) the VCNN method outperformed traditional clustering methods; and 4) the VCNN was on par with PointNet and PointNet++ in classification, and performed the best in segmentation. The proposed method demonstrated LiDAR's ability to separate structural components for crop phenotyping using deep learning, which can be useful for other fields.**

***Index Terms*— Classification, deep learning, LiDAR, phenotype, segmentation, structural components.**

Manuscript received June 10, 2019; revised September 22, 2019; accepted November 7, 2019. Date of publication December 11, 2019; date of current version March 25, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFC0500202, in part by the National Natural Science Foundation of China under Grant 31741016 and Grant 41871332, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA08040107, and in part by the CAS Pioneer Hundred Talents Program. *(Corresponding authors: Yanjun Su; Qinghua Guo.)*

S. Jin, Y. Su, S. Gao, F. Wu, Q. Ma, K. Xu, T. Hu, J. Liu, S. Pang, H. Guan, J. Zhang, and Q. Guo are with the State Key Laboratory of Vegetation and Environmental Change, Institute of Botany, Chinese Academy of Sciences, Beijing 100093, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: jinshichao@ibcas.ac.cn; suyanjun1987@gmail.com; gaos931024@gmail.com; wufangfang@ibcas.ac.cn; maqin@ibcas.ac.cn; kexinxu@ibcas.ac.cn; tianyuhu@ibcas.ac.cn; liujing1030@ibcas.ac.cn; pangshuxin@ibcas.ac.cn; guanhongcan@gmail.com; eve.zhangj@gmail.com; guo.qinghua@gmail.com).

Q. Ma is with the Department of Forestry, Mississippi State University, Starkville, MS 39762 USA (e-mail: qm153@msstate.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2953092

I. Introduction

THE structural components of a plant are usually composed of root, stem, leaf, flower, fruit, and seed [1]. Structural component traits are important indicators for functional analysis [2], which plays a crucial role in phenotyping applications, such as biomass/yield estimation [3], gene discovery [4], and precision agriculture [5]. Accurate, high-throughput, and nondestructive structural components separation is a prerequisite for quantitative phenotyping.

Traditionally, separating phenotypic components (e.g., flower and fruit detection) has been based mainly on image analysis technologies [6]–[8]. These image-based methods were quicker than manual separation, but they could not acquire 3-D information and were sensitive to internal leaf occlusion and external light conditions. Although some stereo images and 3-D reconstruction technologies can obtain 3-D information indirectly, the overlap among different components can affect data accuracy and completeness [9].

Recently, rapid developments in light detection and ranging (LiDAR) have shown its advantages for extracting fine structural 3-D information from vegetation [10]–[13]. A key step for these applications is using automatic methods to extract individual plants and finer target structures [14]–[16]. However, the methods for separating the structural components of crops are very limited [17], [18], and current field phenotyping studies focus on the individual and group level [18]–[20], [21]–[23]. There is an urgent need to develop new methods for separating structural components for phenotyping using LiDAR.

After reviewing the methods for separating plant components, we found that current methods have mainly been developed in forestry for separating the branches (stems) and leaves of a tree. These methods can be summarized into three types: threshold-based (e.g., intensity, multiwavelength, and waveform threshold), geometry-based (e.g., point-based), and machine-learning-based methods. The threshold-based methods rely on the threshold of intensity or waveform width [24], [25] in the segmentation. Threshold-based methods are empirical and less popular because they often require multiband or full waveform LiDAR systems. Moreover, their application in crop studies is limited because the optical properties of crop stems and leaves are too similar to be discriminated through thresholds. Geometry-based methods, such as the point-based method, can separate

different types of objects through their geometric features [14]. However, it is hard to define an optimal geometric feature for manifold and irregularly shaped crop leaves. By contrast, machine-learning-based methods, especially the recently developed deep-learning-based methods, are tailored to learn complex features from large amounts of high-dimension data automatically [26]. These have the potential to outperform traditional methods [27].

Convolutional neural networks (CNNs), which are one of the most popular deep learning architectures, have become the state-of-the-art methods in object detection [28], classification [29], and segmentation [30]. In the field of plant phenomics, image-based CNNs have been successfully used for fruit and stress (e.g., disease) detection [31]–[34], species classification [35], and rice panicles segmentation [36]. Recently, some 3-D object-oriented CNNs have been developed, such as the voxel-based methods [30], [37], octree-based methods [38], multisurface-based methods [39], multiview-based methods [40], and point-cloud-based methods [41], [42]. The voxel-based methods can effectively preserve the spatial relationship between voxels, which is promising for segmenting closely connected objects.

Despite the advantages of LiDAR data and deep learning methods, there are some challenges when combining these two into applications. In this research, there were three challenges.

1) *Lack of a Benchmark Data Set:* Some publicly available data sets are available for 2-D[1] and 3-D[2] semantic segmentation of land cover, but there are no benchmark data for structural component separation in the field of phenotyping.
2) *Unstructured Point Data Require a Specifically Defined CNN:* Current CNNs (e.g., UNet [43] and Mask R-CNN [44]) can handle structured data well, such as 2-D grid images. However, CNNs for point clouds with only an unstructured and unordered format have not been fully exploited.
3) *Limited Information in Point Clouds:* When applying CNNs for remote sensing image analysis, both geometry (2-D gridded coordinate) and spectral information can be used. However, LiDAR points usually contain basic geometry (3-D unordered coordinate) information, while other attributes (e.g., intensity, and return numbers) are rarely recorded and used. It is hard to use point classification without spectral information.

In this study, we provided solutions to these challenges by proposing a multitask voxel-based CNN (VCNN) to classify and segment the stems and leaves of individual maize plants from terrestrial LiDAR data. In preparing training and testing data, every point of the LiDAR data was labeled as stem or leaf semiautomatically, and the points of the maize plant with labeled classes were voxelized as inputs to the VCNN. Through a hierarchical structural (encoder–decoder) feature extraction and residual learning method, we demonstrated the use of a weighted cross entropy and a discriminative loss function for classification and segmentation, respectively. The multitask VCNN was trained "end-to-end" using a joint loss (i.e., sum of classification and segmentation losses) function and was accelerated by graphics processing unit (GPU) computing, which enabled more efficient processing. The main contributions of our methods covered four aspects. First, we introduced a new objective function (i.e., discriminative loss) for segmentation, which operates at the voxel level in feature space. Second, the segmentation loss and the classification loss were combined during training so that the two tasks can contribute to each other as in many multitask learning neural networks. Third, using the fully CNN to extract features in an "end-to-end" process we can apply input data of various sizes and predict output classes at the voxel level. Fourth, using the "encode–decode" architecture, similar to U-Net, can concatenate hierarchical features, including semantic and location features. These connections can also improve the convergence time of the model. Moreover, in each stage of encoding (down-sampling) and decoding (up-sampling), the input was treated in a residual learning way (ResNet), and therefore, improved both model convergence time and performance even with small training samples.

To provide a clear illustration of the method, the rest of the article is organized as follows. In Section II, the architecture of the network and loss function is described. In Section III, details of the experimental analyses are specified, including study area and data collection, training and testing data preparation, network training and testing, accuracy assessment, and comparison experiments. Section IV presents the results of the experiments. Then, Section V provides the discussion of the performance of our method and the differences with other methods. Finally, conclusions are drawn in Section VI.

II. Methodology

The VCNN was an individual maize-oriented method for maize component (i.e., stem and leaf) classification and segmentation. To retain the 3-D geometric correlations among LiDAR points and let 3-D convolutional operations become possible, we converted the original LiDAR point clouds to voxels, and used the voxels as the input of VCNN. The state of the art fully convolution network, encoder–decoder architecture and residual learning (ResNet) used in 2-D CNNs were adopted here to extract hierarchical features in an "end-to-end" way [30], [45], [46]. Moreover, a multitask joint learning with a discriminative loss was used to achieve both stem and leaf classification and segmentation. As Fig. 1 shows, the down-sampling (encoder) process included several stages, each of which consisted of several convolution, activation, and pooling layers connected through residual learning (i.e., skip connection), which extracted features from low to high levels. The up-sampling (decoder) process was completed by combining high-level features in the current state with low-level features generated previously. Finally, the neural network generated a 2-channel volumetric feature map for classification through a softmax function. A 34-channel feature map was used to segment instances of stems and leaves using a mean-shift algorithm, which can find clusters by iteratively updating

[1]http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html
[2]http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html

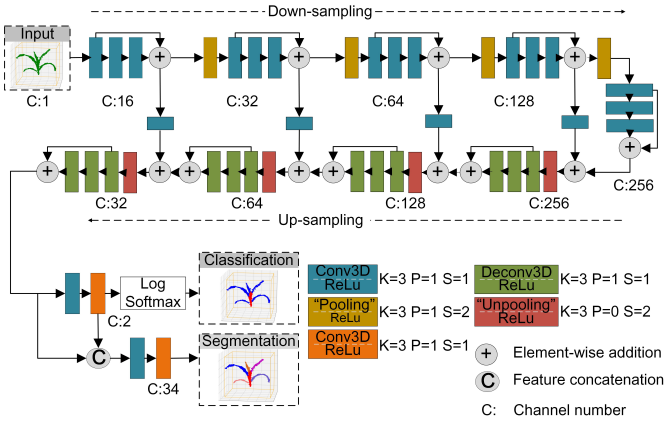


Fig. 1. Architecture of the VCNN. The input layer is a normalized and voxelized maize sample, shown in green. (Top) Hidden layer containing down sampling and (Bottom) Up-sampling stages which both comprise 3-D convolution (Conv3D), activation (ReLU), and pooling operations. The outputs are the classification voxels in blue and red, and segmentation voxels in different colors.

candidates for cluster centers to be the mean of points within a given neighborhood. The mean-shift method is nonparametric and has proved to be insensitive to noise and useful for clustering arbitrary shapes in a complex multidimensional feature space [47]. The details of the methods are as follows.

A. Voxel-Based Convolutional Neural Network

The VCNN is a type of CNN whose input data are organized as voxels. The layers of a CNN can be categorized into three main types: the convolutional layer, max pooling layer, and fully connected layer. The convolutional layer is composed of many filters with learnable weights and biases. Each filter was used to slide over the input data with an appropriate perception field [i.e., kernel size (K), stride (S), and padding (P)], which gives the convolution value (i.e., dot product between filter and input data within the filter). The convolution value was activated with a nonlinear function [e.g., sigmoid, hyperbolic tangent, and rectified linear unit (ReLU)] to get different features (e.g., edge features) and passed to the next layer. Generally, the next layer was a pooling layer, which performed a down-sampling operation by reducing the spatial size of the representation [48], [49]. The most frequently used pooling operation was max pooling [50], which used the maximum value of a 2×2 filter. Finally, the fully connected layer computed the probability scores of belonging to each class [35]. In this way, the VCNN transformed the original data (pixel values) layer by layer into the final class scores, which were used to classify and segment instances. In this study, to achieve voxel level segmentation, the pooling and fully connected layers were both substituted by convolutional operations in the down-sampling and up-sampling stages.

The down-sampling process had five stages. Each stage comprised three convolutional layers with an activation function, followed by a pooling layer. The convolutional layers used a volumetric kernel of a $3 \times 3 \times 3$ voxels filter to slide over the input data with appropriate stride and padding, and the convolution layers were connected using the residual learning method (Fig. 1). After convolution, a nonlinear ReLU activation function was applied to the feature maps. The activated feature maps were then down-sampled to produce higher level features by pooling layers. In this study, the pooling layer was a convolutional layer using a volumetric kernel size of $2 \times 2 \times 2$ voxels and a stride of two voxel sizes. Therefore, the resolution of each stage was halved, which had the same effect as the pooling layer in a typical CNN. The feature maps of each stage were sent to the next stage. For example, in the upper part of Fig. 1, the input data was 4-D (i.e., $C \times L \times W \times H$). C denoted the number of channels (features), which was the class label of each voxel determined by the majority class of its internal points. L, W, and H were the number of the voxels in three directions of the voxelized data. If the input size was $1 \times 93 \times 32 \times 67$, after using 16 3-D convolutional kernels of $3 \times 3 \times 3$ voxels in size in the convolutional layer with ReLU activation function, we got a feature map with 16 channels ($16 \times 93 \times 32 \times 67$). Then, a pooling operation with 32 filters doubled the features and halved the resolution, which meant that the size of feature maps after pooling was $32 \times 47 \times 16 \times 34$. The process (i.e., convolution, activation, and pooling) was repeated five times in this study, which was also known as the depth of the network.

In the lower part of the network, the up-sampling process comprises several deconvolutional layers. Each deconvolutional layer comprised one 3-D transposed convolution and three convolutional layers with an ReLU activation function, enabling up-sampling with features of different levels [30]. The input of each deconvolutional layer combined the current feature map of the higher level and the prior feature map of the lower level created by down-sampling (Fig. 1). The 3-D transposed convolution doubled the spatial resolution of the input data (higher level), which was then combined with the lower level feature from the down-sampling stage. The combined features were processed with three convolutional operations connected through residual learning to learn more information. Finally, we achieved a feature map of 32 channels, which was used for further classification and segmentation. On the upper part of the branch, the feature map was used directly to do classification. A convolutional layer with a volumetric kernel of $1 \times 1 \times 1$ was used to get a 2-channel feature map, whose size was the same as the input data. The 2-channel feature map was converted into the probability of foreground and background using a softmax function and was labeled with a class according to the max probability. The labeled voxels were used to calculate the classification loss with ground truth. After classification, we used a multitask learning method for segmentation [51]. We concatenated the 32-channel deconvolutional feature map and the 2-channel classification feature map to get a 34-channel feature map. The connected 34-channel feature map provided enough information for segmentation using a mean-shift clustering method and was further used to get the segmentation loss. To make the process more readable, we visualized the extracted features at different hierarchical levels after each down sampling (pooling), up sampling (unpooling), and features for classification and segmentation (Fig. 11).

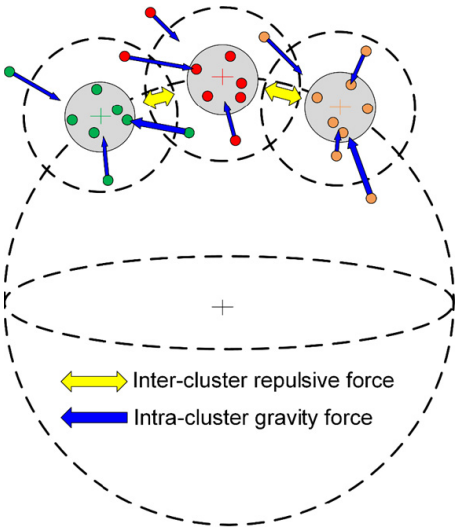


Fig. 2. Diagram of the stem and leaf instances segmentation loss. The intercluster repulsive force can make the distances between different instances tend to be larger and the intracluster gravity force makes the distance between the same instance tend to be smaller. Different colors of points represent different clusters.

B. Loss Function for Classification and Segmentation

For stem and leaf classification, the distance between the predicted class from our model and the ground truth were expected to be close. The cross-entropy loss can accurately describe the distance between the trained model and the ideal model using the empirical distribution and the predicted distribution [52], which is ubiquitous in many classification approaches. In this study, the number of points of the stem class was less than the leaf class. To solve the data imbalance problem, we used a weighted cross-entropy loss (L_c), which is described as follows:

$$L_c = -\sum_i w_i y_i' \log y_i \tag{1}$$

where i represents different predictions, w_i is the weight of class i, which is calculated by subtracting the ratio of the class from 1; y_i is the predicted value (probability of each class), y_i' is the target value (one hot encoding).

For segmentation, we used a discriminative loss function based on a previous study on distance metric learning, which achieved a competitive performance on tasks relating to image-based traffic scene segmentation and leaf instance segmentation [53], [54]. The loss function encouraged the network to cluster points of the same instance and separate different instances with a wide margin in feature space.

The segmentation loss used two competing forces to minimize segmentation errors (Fig. 2). One was the intracluster gravity force, which penalized large distances to cluster points with the same label. Points within one cluster were used to compute the distance to the cluster center and get the mean value of all clusters (2). The other was intercluster repulsive force, which penalized small distances to segment points of clusters with different labels (3). A regularization loss was used to pull all clusters toward the origin as well as to prevent overfitting (4). Thus, the regression loss function comprised three parts (5), the intravariance loss, interdistance loss, and regularization loss

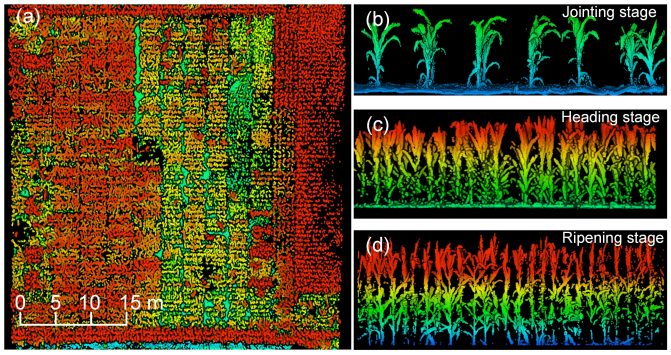


Fig. 3. Examples of (a) registered terrestrial LiDAR point cloud of the maize field collected in June 2017, and LiDAR point cloud sample at the (b) jointing stage (collected in June 2017). (c) Heading stage (collected in July 2017). (d) Ripening stage (collected in August 2017).

$$L_{\text{var}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_C}[\|x_i - \mu_C\|_2 - \delta_v]_+^2 \tag{2}$$

$$L_{\text{dist}} = \left\{\frac{1}{C(C-1)}\sum_{C_A=1}^{C}\sum_{C_B=1}^{C} \times [2\delta_d - \|\mu_{C_A} - \mu_{C_B}\|_2]_+^2, C_A \neq C_B\right\} \tag{3}$$

$$L_{\text{reg}} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\|_1 \tag{4}$$

$$L_r = \alpha \cdot L_{\text{vat}} + \beta \cdot L_{\text{dist}} + \gamma \cdot L_{\text{reg}} \tag{5}$$

where L_{var} is the loss of intravariance of a single segmented instance; C is the number of segmented instances; N_c is the number of points of instance C; μ_c is the center of instance C, x_i is a point belonging to C; δ_v is a given threshold of inner variance; $\|\cdot\|_1$ is L1 distance; $\|\cdot\|_2$ is L2 distance; $[x]_+$ is the maximum value between x and 0; L_{dist} is the loss of interdifference of two adjacent objects; μ_{C_A} is the center of object C_A; μ_{C_B} is the center of object C_B; δ_d is a given distance between center of C_A and center of C_B; L_{reg} is the regularization loss; and α, β, and γ are given parameters. In this study, we set $\alpha = 1$, $\beta = 1$, $\gamma = 0.001$, $\delta_v = 1$, and $\delta_d = 3$.

The final loss function of the VCNN is defined as follows:

$$L = L_c + L_r. \tag{6}$$

III. Experimental Design

A. Study Area and Data Collection

The study area was at the China Agricultural University, Beijing, China, with an area of 1300 m^2 [Fig. 3(a)]. Maize (*Zea mays* L.) individuals were planted on May 17, 2017, at intervals of around 0.5 m. LiDAR data were acquired using a FARO Focus^{3D} X330 HDR scanner in June, July, and August, 2017, when the maize was at the jointing, heading, and ripening stages, respectively [Fig. 3(b)–(d)]. The scanner was

TABLE I
SPECIFICATIONS OF THE LiDAR SCANNER USED IN THIS STUDY

Sensor	FARO Focus^{3D} X 330 HDR
Laser wavelength (nm)	1550
Laser beam divergence (mrad)	0.19
Field of view (°)	Horizontal: 360°; Vertical: 300°
Angular resolution (°)	Horizontal: 0.009°; Vertical: 0.009°
Detection range (m)	0.6m - 130m indoor or outdoor with upright incidence to a 90% reflective surface
Pulse rate (kHz)	244
Maximum scanning rate (Hz)	97
Distance accuracy	3mm @10m @90% reflectance
Scanner weight (kg)	5.2
Dimensions (mm)	240 x 200 x 100
Laser class	Laser class 1
Beam diameter at exit (mm)	2.25

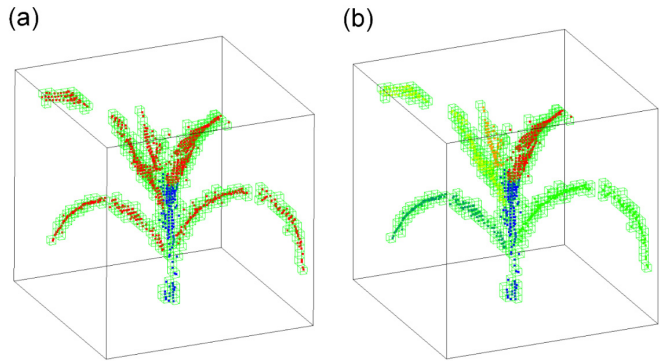


Fig. 4. Examples of the normalized and voxelized individual maize samples for classification and segmentation. (a) Training example for classification, whose stem is in blue and all leaves are in red. (b) Training example for segmentation, whose stem is in blue and all other individual leaf instances are in different colors.

mounted on a tripod and operated at a pulse rate of 244 kHz. The scanner had a maximum scanning rate of 97 Hz and large field of view (horizontal: 360°; vertical: 300°), which facilitated good scanning accuracy (0.003 m at 10 m at 90% reflectance) and angular accuracy (horizontal: 0.009°; vertical: 0.009°) (Table I).

In this study, ten scans were collected evenly distributed within the study area to ensure data quality. Through registration in FARO Scene software, thousands of samples at each growing stage were acquired with a registration accuracy of 0.002 m for the same point from different scans (Fig. 3).

B. Training Data Preparation

To prepare data for stem-leaf classification, we labeled the points of stems and leaves with different classes (0 for stems and 1 for leaves) using a semiautomatic method [Fig. 4(a)]. For segmentation, the points of different leaf instances were further labeled as different classes [Fig. 4(b)]. The semiautomatic methods contain two steps. First, individual maize samples covering different sizes, shapes, point densities, and leaf numbers (Table II) from different growth stages were segmented automatically through a deep-learning-based method [18] and then manually refined using the Green Valley International LiDAR360 software. Second, each refined individual maize plant was further segmented into stem and leaf instances with the help of an automatic region growing algorithm [17]. The segmented stem and leaf instances were labeled with targeted classes.

We prepared 423 ground truth samples. A subset of 320 samples from all ground truth samples were used to generate 3000 training samples through data augmentation to reduce the manual workload as well as to keep the balance of data distribution in three growing stages. For each ground truth sample, which had been segmented into stem and leaf instances, we rotated every segmented leaf instance with the stem at a random angle (0°–180°) and added a small vertical shift (1%–10% of the stem height) to the leaf-stem node location. This process was repeated several times (5–15 times) for each maize plant at different growing stages. Finally, we checked the repeated results visually, ending up with 3000 (including the initial 320 samples) maize samples. These training samples covered heights varying from 0.13 to 2.49 m and crown diameters ranging from 0.11 to 1.58 m. The corresponding ratio of crown diameter to plant height (CHR) ranged from 0.50 to 4.04, which is a good representation of the compactness of an individual maize plant [17]. The leaf numbers of test samples varied from 2 to 13, and the point density varied from 19 023 to 1 177 462 pts/m^2 (Table II).

For better model training and generalization, each training sample was normalized before voxelization, as follows:

$$(x, y, z) = \frac{(x, y, z) - \min(x, y, z)}{\max(x, y, z) - \min(x, y, z)} \tag{7}$$

where x, y, and z are three n by 1-D matrix that constitute the 3-D point coordinates of each sample, and n is the number of points in each sample.

The normalized data were voxelized to reduce the size of the input data as well as keep the 3-D architecture. In this study, each maize plant was divided into $L \times W \times H$ voxels. L, W, and H were the number of the voxels in length, width, and height directions. Similar to normalization, the shortest edge of the three directions was guaranteed to have 32 voxels (the edge length of each voxel was around 0.004 m) using (8)

$$\begin{aligned} \text{voxel} &= \frac{\min(\text{width}, \text{length}, \text{height})}{32} \\ L &= \frac{\text{length}}{\text{voxel}} \\ H &= \frac{\text{height}}{\text{voxel}} \\ W &= \frac{\text{width}}{\text{voxel}} \end{aligned} \tag{8}$$

where length, width, and height are the edge length of the maize bounding box in the directions of length, width, and height, voxel is the resolution of a voxel (unit is m), which is calculated by dividing the minimum of length, width, and height by 32.

After voxelization, a voxel may contain zero, one, or more than one point(s). Only voxels containing one or more than one point(s) were labeled and used as training samples. If a voxel had one point, the class of the voxel was determined by the class of the point. If a voxel had more than one points,

TABLE II
STATISTICS OF THE HEIGHT, CROWN DIAMETER, POINT DENSITY, CHR AND LEAF NUMBER OF MAIZE INDIVIDUALS USED AS TRAINING AND TESTING SAMPLES

	Training data				Testing data			
Properties	Max	Min	Mean	SD	Max	Min	Mean	SD
Height, m	2.49	0.13	0.98	0.54	2.61	0.16	1.03	0.61
Crown diameter, m	1.58	0.11	0.60	0.26	1.49	0.12	0.59	0.30
Crown height ratio (CHR)	4.04	0.50	1.61	0.68	1.55	0.20	0.70	0.32
Point density, pts/m^2	1177462	19023	177747	145560	923192	23475	123543	130106
Leaf number	13.00	2.00	7.07	1.98	12.00	3.00	7.18	1.92

the class of the voxel was determined by the majority class of the points. If there was more than one majority class (0 and 1) in a voxel, the class of the voxel was randomly selected.

C. Testing Data Preparation

As mentioned in Section III-B, the remaining 103 samples (33, 33, and 37 samples from June, July, and August, respectively) of the total 423 ground truth samples were used for testing the performance of the proposed algorithm. These samples covered heights ranging from 0.16 to 2.61 m, crown diameters ranging from 0.12 to 1.49 m, and CHRs ranging from 0.20 to 1.55. The leaf numbers of testing samples varied from 3 to 12, and the point density varied from 23 475 to 923 192 pts/m^2 (Table II). Testing samples were classified into stem and leaves as well as segmented into stem and leaf instances using the same procedure as for the training samples. These classified testing samples were also normalized and voxelized using the same methods used to prepare training data.

D. Network Training

The VCNN was trained "end-to-end" using the back propagation algorithm [55] and the PyTorch framework (http://pytorch.org/) with a strong GPU (GIGABYTE GTX 1080 WF3OC) acceleration. The 3000 training samples were randomly picked and added to a rotation transformation, and then sent to the network in each epoch. Through this "data augmentation," each training epoch was unique, which enhanced the model performance and avoided overfitting. In each epoch, the batch size was 1, because the sizes of each voxelized sample were not the same. We used vanilla gradient descent to optimize the loss function (L) with a time-dependent learning rate. In this study, the initial learning rate was 0.0001 and was divided by the square root of the epoch at every 20 epochs. It remained unchanged after a hundred epochs. The model was trained day and night until the training loss was satisfied (generally smaller than 0.01) and almost unchanged (i.e., the model converged).

E. Network Testing

The testing process used the same network obtained from the training process and generated the voxel-based classification and segmentation results. The classification result was fixed at two classes (0 represents stem and 1 represents leaves), but the number of segmented instances was unknown. In this study, the mean-shift algorithm was used to cluster different instances with a 34-channel (i.e., $2 + 32$) feature map. The search radius of the mean-shift algorithm was the trained parameter δ_d. The automatically clustered result may have some small instances with only a few points, which were refined using the same methods as our previous work [17]. If the number of points of an instance was less than a given threshold n, these points were reallocated to their nearest instance. In this study, n was set to a fixed value of 30, which enabled batch testing of all 103 samples.

To get the class of each point, we matched the voxel level outputs with the initial points. In this study, a point class was determined by the class of the voxel it was located in.

F. Accuracy Assessment

The instance-to-instance assessment can give a general and direct result of the model performance, and the point-to-point accuracy assessment can give a more objective and precise comparison. The classification accuracy at both instance level and point level were represented by simple classification accuracy function, which was defined as follows:

$$\text{Overall accuracy} = \frac{\sum \text{TP}}{N} \tag{9}$$

where TP is the number of instances at instance level or points at a point level that have the same label as ground truth, and N is the number of total instances or points of each maize plant.

The segmentation accuracy was conducted for each stem and individual leaf predicted by our network with ground truth at both instance level and point level. If an instance or a point was labeled and segmented as the same class, it was a true positive (TP); if an instance or a point was mis-segmented, it was a false negative (FN); if an instance or a point label did not exist but was segmented from instances or points, it was a false positive (FP). We expected higher TP, lower FN, and lower FP to get higher accuracy. We also calculated the recall (r), precision (p), and F-score (F) for stem and leaf instances or points using (10) [56] to evaluate the

segmentation accuracy

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + FN}$$
$$p = \frac{\mathrm{TP}}{\mathrm{TP} + FP}$$
$$F = 2 * \frac{r * p}{r + p}. \tag{10}$$

G. Comparison With Other Methods

As mentioned above, intensity-based methods are not applicable for classifying crop stem and leaves owing to their similar optical properties. In this study, the VCNN method was compared with a geometry-based method known as median normalized-vector growth (MNVG) [17]. In addition, considering that the classification and segmentation are essentially clustering problems, the VCNN was compared with the most popular machine-learning clustering methods, including unsupervised density-based spatial clustering of applications with noise (DBSCAN) [57] and K-means [58]. To assess our method against similar CNNs, two of the state-of-the-art CNNs, PointNet and PointNet++, were also compared [41], [42].

The MNVG is a geometry-based segmentation method that uses a regional growth algorithm to grow stem and leaf. Two parameters need to be set. One is the radius of the search neighborhood (R) in the growth stage, the other is the minimum number (n) of points that are needed to form a leaf in the post processing stage. In this study, R was set to an optimized value using a gradient search method, and n was set to the same value (30) as used in VCNN method for fairer comparison.

The DBSCAN method can cluster points of an arbitrary shape based on their density, which needs two parameters to be defined, that is, the minimum number of points required to form a cluster (*MinPts*) and the radius of the neighborhood of each point (*Eps*). In this study, *MinPts* and *Eps* were set at the optimal value using a grid search method. The K-means method can partition points into K clusters in which each point belongs to the cluster center with the nearest distance. In this study, parameter K for the K-means method was set according to the total numbers of leaf and stem instances in ground truth samples. Both the DBSCAN and the K-means method were carried out with the *sklearn* package in Python3 language [59].

PointNet and PointNet++ are two point-based deep learning methods, and the latter is an improved version of the former that can learn more local features. Both the two methods were designed to do object classification, semantic segmentation, and instance segmentation from points. However, the semantic/instance segmentations need to define an output number of semantic/instance classes; this limits their direct applications for stem-leaf instance segmentation because the leaf number of individual maize is unknown before being segmented. Therefore, we first used PointNet/PointNet++ to do stem-leaf semantic segmentation as the classification task of VCNN. Then, the trained feature of the last layer of the semantic network was used to do instance segmentation using the same clustering method as the segmentation task of VCNN. In the clustering and optimization stage, two parameters are needed, as used in VCNN method. Parameter R was set to an optimized value using a gradient search method, and n was set to the same value (30) as used in VCNN method for fairer comparison.

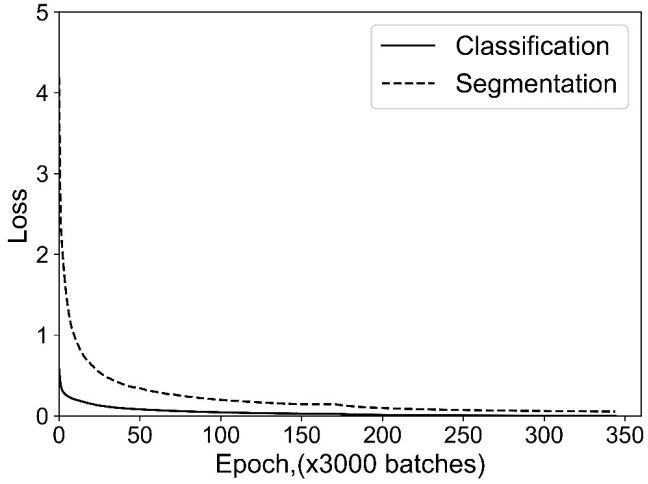


Fig. 5. Average training loss of stem-leaf classification and segmentation in the total 350 epochs.

The DBSCAN, K-means, MNVG, PointNet, and PointNet++ methods were used to segment (cluster) the stem and leaf instances in all the 103 testing samples. The segmentation results were compared with ground truth. The nearest cluster to the stem in ground truth data was treated as segmented stem and other segmented clusters were treated as leaf instances to derive the classification results. Classification and segmentation results were evaluated using the overall accuracy (OA) and F-score at the point level, respectively.

IV. Results

A. Results for Training Loss

In this study, the VCNN was trained in 350 epochs in total. Each epoch contained 3000 batches with a batch size of 1. The training loss decreased quickly for the first one hundred epochs (Fig. 5). After that, the decrease rate was relatively slow. The final losses for classification and segmentation were 0.002 and 0.055, respectively. The total time of the training was about 48 h on a PC with intel i7 CPU, 16 GB RAM, and a NIVIDA GTX 1080 GPU.

B. Results for Stem-Leaf Classification

The classification results for the 103 maize plants were assessed both visually and quantitatively. Fig. 6 shows some typical results according to different properties (i.e., maize height, CHR, leaf number, and point density) and the highest (lowest) accuracy cases (Fig. 6).

Visual assessment suggested that the predicted results and the ground truth data were very similar in different cases. However, we still found some subtle misclassifications: 1) in Fig. 6(f), the points of a broken leaf are misclassified

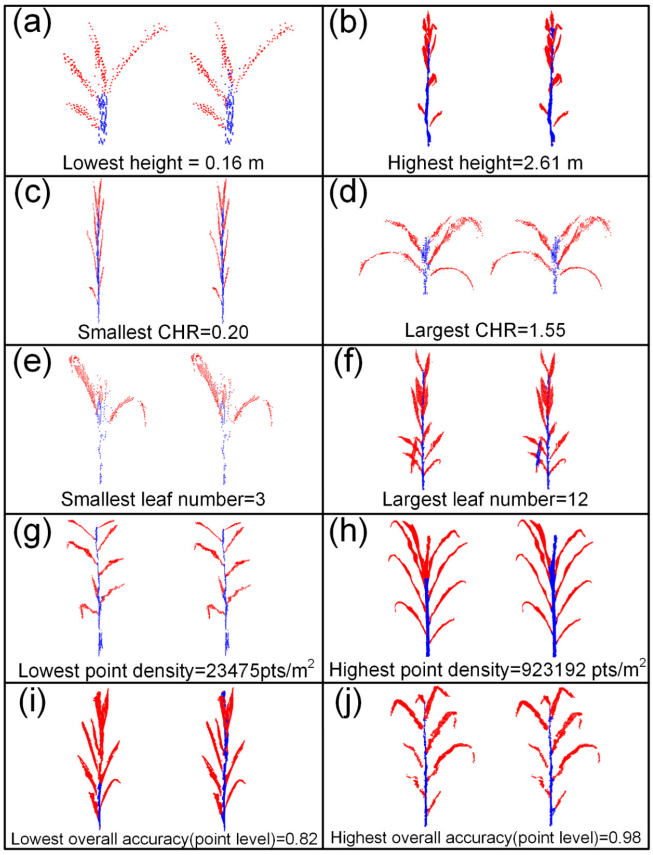


Fig. 6. Classification results for the 103 selected samples using the VCNN method. In each subplot, (left and right) ground truthed by manual classification and the VCNN classified results, respectively. Points of stem and leaves are in blue and red, respectively. (a)–(j) were selected due to their representativeness of different properties annotated in the subfigures.

as stems; 2) the relatively young and small leaves [top center leaves in Fig. 6(h) and (i)] are misclassified as stems. The quantitative evaluation results showed that: 1) the mean, minimum, and maximum classification accuracy were all 1.0 at the instance level; and 2) the mean classification accuracy was 0.91 at the point level, with a maximum and minimum accuracy of 0.99 and 0.76, respectively (Table III).

C. Results for Stem-Leaf Segmentation

The segmentation results of the 103 maize plants were also assessed both visually and quantitatively. Fig. 7 shows some typical results according to different properties (i.e., maize height, CHR, leaf number, and point density) and the highest (lowest) F-score cases.

Through visual assessment, the predicted results and the ground truth data were also very similar in different cases. However, some regular mis-segmentation was found. 1) A few instances (points) are misclassified as leaves shown in the black circle [Fig. 7(b) and (c); 2) the top center [Fig. 7(e) and (i)] and broken leaves [Fig. 7(f)] are clustered as stems; and 3) the top center leaf of the lowest F-score case is also partly misclassified. The quantitative evaluation results showed that: 1) at the instance level, the mean value of r, p, and F were 0.97, 0.96, and 0.96, respectively;

TABLE III

ACCURACY ASSESSMENTS OF THE STEM-LEAF CLASSIFICATION (OA) AND SEGMENTATION (RECALL/r, PRECISION/p, AND F-SCORE/F) AT INSTANCE AND POINT LEVELS USING THE 103 TEST SAMPLES

	Instance level				Point level			
Statistics	OA	r	p	F	OA	r	p	F
Max	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.97
Min	1.00	0.75	0.64	0.74	0.76	0.62	0.81	0.63
Mean	1.00	0.97	0.96	0.96	0.91	0.89	0.93	0.89
SD	0.00	0.06	0.08	0.05	0.05	0.08	0.04	0.07

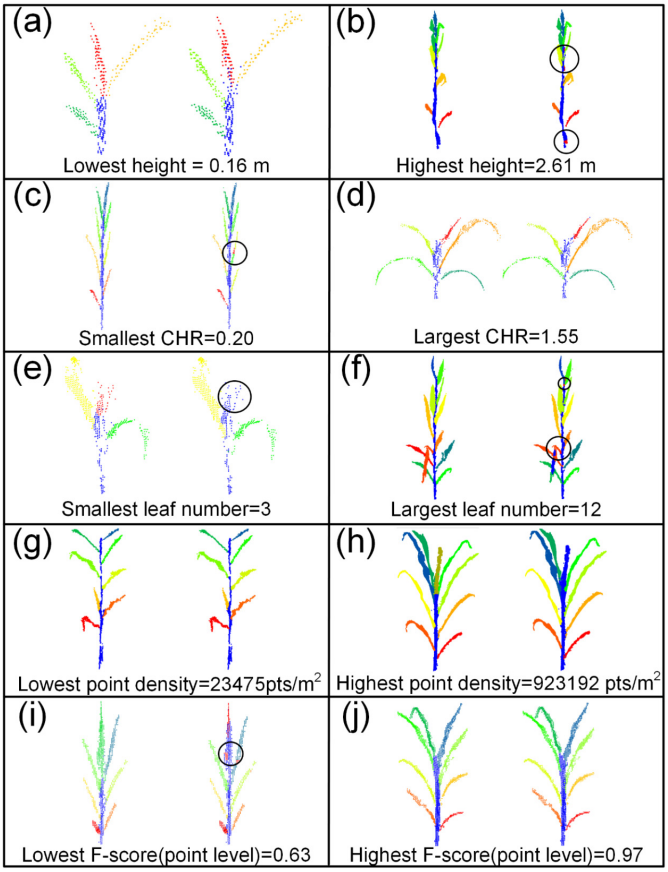


Fig. 7. Segmentation results for the 103 selected samples using the VCNN method. In each subplot, (left and right) ground truthed by manual segmentation and the VCNN segmented results, respectively. Points of stem and each leaf instance are in different colors. (a)–(j) were selected due to their representativeness of different properties annotated in the subfigures.

2) at the point level, the mean r, p, and F were 0.89, 0.93, and 0.89, respectively; and 3) the standard deviations of r, p, and F were lower than 0.08 at both instance and point level. Moreover, we found the mean accuracies of classification and segmentation were almost at the same level without a trend of change at different growth stages (Fig. 8).

D. Comparison With Other Methods

In terms of classification, the OA of DBSCAN, K-means, MNVG, PointNet, PointNet++, and VCNN were 0.528,

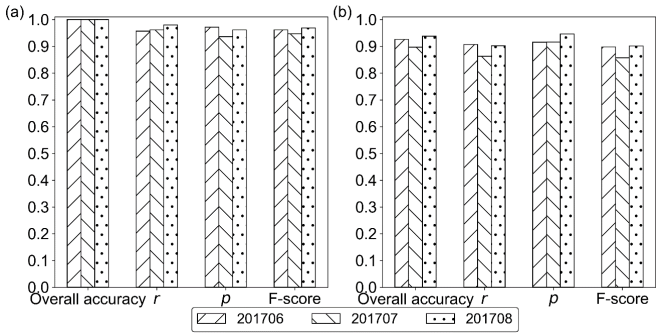


Fig. 8. Mean OA, recall (r), precision (p), and F-score of the three growth stages at both (a) instance and (b) point levels.

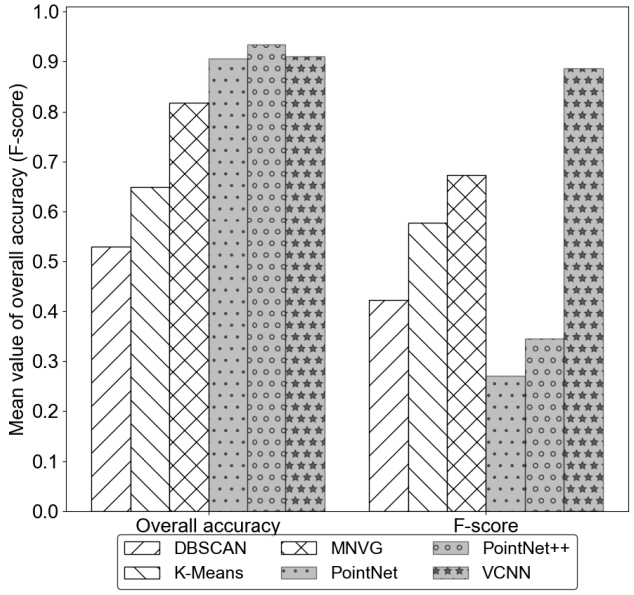


Fig. 9. Accuracy comparison of DBSCAN, K-means, MNVG, PointNet, PointNet++, and VCNN methods in stem-leaf classification (OA) and segmentation (F-score) at the point level.

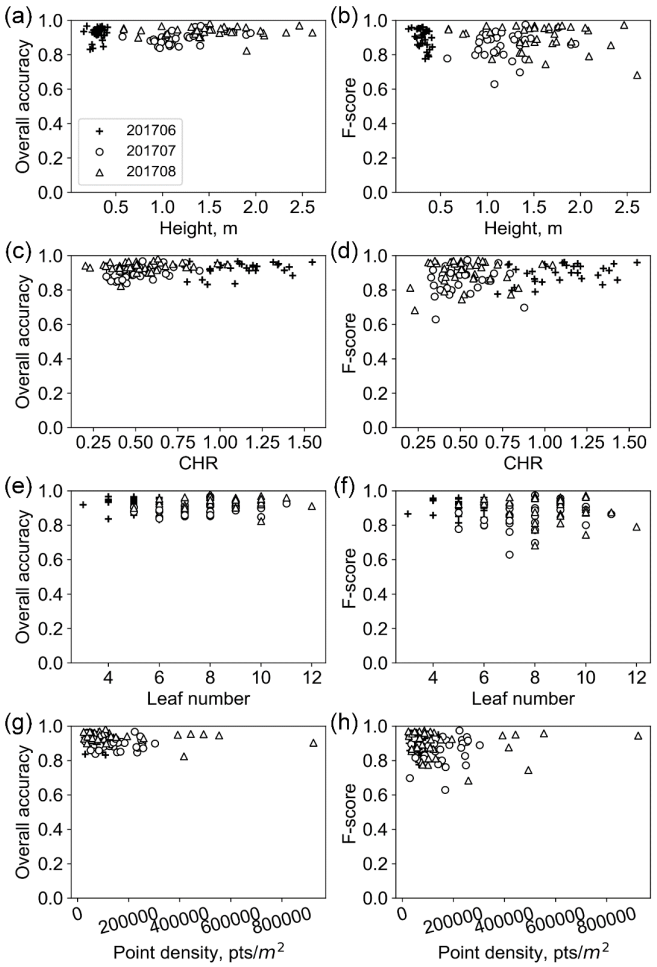


Fig. 10. Influences of different properties on the accuracy of classification (OA) and segmentation (F-score) of the VCNN method in the three growing stages at the point level. (a) and (b) Results of height. (c) and (d) Results of CHR. (e) and (f) Results of leaf number. (g) and (h) Results of point density.

0.650, 0.818, 0.906, 0.934, and 0.911, respectively. Therefore, the VCNN was better than all heuristic methods (DBSCAN, K-means, and MNVG) and on par with the state-of-the-art deep learning methods. In terms of segmentation, the F-scores of DBSCAN, K-means, MNVG, PointNet, PointNet++, and VCNN were 0.42, 0.57, 0.67, 0.27, 0.35, and 0.89, respectively. Therefore, the VCNN performed the best of all the segmentation methods, followed by MNVG, K-means, DBSCAN, and the two other deep learning methods. In addition, the stem-leaf classification accuracy of each of the six methods was higher than segmentation accuracy (i.e., F-Score) (Fig. 9).

V. Discussion

A. Stem-Leaf Classification

The VCNN achieved an overall classification accuracy as high as 1.0 at the instance level for different types of maize plants (Table III, Fig. 8). However, the prediction was different from the ground truth at the point level. The lowest classification accuracy (0.82) appeared in Fig. 6(i), which might be mainly caused by the over classification of a stem that covers a top center leaf growing vertically along the stem. By contrast, the highest classification accuracy (0.98) appeared at Fig. 6(j). The reason for the high accuracy may be that the leaves and stem did not overlap, and there were no damaged or vertically growing leaves. The occlusion and overlapping of leaves and stems can be represented by height, CHR, leaf number, and point density to a certain extent. Theoretically speaking, a maize plant with higher height, smaller CHR, more leaves, and smaller point density tends to be more occluded/overlapped and shows a lower accuracy [17]. However, the OA and F-score were almost steady across all the test samples. Overall, the VCNN method was robust to different heights, CHRs, leaf numbers, and point densities (Fig. 10).

In addition, the classification prediction was "end-to-end" and nonparametric, which is more convenient and efficient [60], [61]. Compared with the traditional threshold-based classification method [24], [25], the VCNN method does not depend on the information of intensity and waveform, which reduces the requirements for hardware and data quality. Compared with the traditional geometry-based classification

method [14], [17], the proposed VCNN method classifies and segments stems and leaves directly based on original 3-D point locations and self-learned features rather than a human-defined geometry rule, which will be easier to generalize for different applications.

B. Stem-Leaf Segmentation

The segmentation results for different maize samples showed high accuracy in terms of r, p, and F, whose mean values were all more than 0.95 and around 0.9 at instance and point levels, respectively. The reason is that the segmentation used a high dimensional feature map (e.g., the well trained 34-D feature map), which contained different levels of information. In addition, the classification accuracy guaranteed the segmentation accuracy to some extent, because "end-to-end" training with a joint loss can maximize overall performance [62], [63]. The 34-D features contained the 2-D features for classification, which were a reliable basis for improving segmentation accuracy. The lowest F-score (0.63) appeared in Fig. 7(i), which was mainly because of the mis-segmentation of the vertically growing leaf at the top center. Although this problem is common, the VCNN method can segment these types of plants correctly in some situations [Fig 7(b)]. Theoretically, if leaf instances are distributed evenly and have no overlap or vertical growth leaf the F-score will be higher, like the example in Fig.7(j). The accuracy of stem-leaf segmentation was lower than for classification because there were more classes to be predicted and the number of classes was uncertain in the segmentation task.

There were two parameters for segmentation. One was the search radius (δ_d) of the mean shift, and the other was the smallest number of leave (n) for segmentation refinement. In this study, the search radius was the distance between two different instances, which was optimized to fit the current model because it was trained using model training. Parameter n is the most important factor, which should be set according to the number of leaves for each sample. The influence of parameter n has been discussed in our previous work [17], which gives more details about how to set the parameter. In this study, we analyzed the influence of maize height, CHR, leaf number, and point density on the segmentation accuracy at the point level. Our results prove that the VCNN performed well with different data qualities and growth stages. The robustness may come from two aspects. First, the training/testing samples with different point densities were generated by randomly sampling points from the very complete and high-density LiDAR datasheets. Second, the VCNN was a voxel-based method, which has a low requirement of point density. As long as a voxel has one point, the method can perform as well as a voxel with several points. However, we admit that if a LiDAR data set was collected with different setups (e.g., viewing angles), the results might be different. Further studies are still needed to evaluate the extensibility of the developed VCNN model. Nevertheless, even if the current developed model cannot be directly used to other data sets with different characteristics. We believe that by providing enough training samples, the proposed VCNN method can ultimately generate a universal model for maize stem/leaf classification and segmentation.

The reasons why we segment leaves without removing the stem first are for two aspects.

1) We found the segmentation results of leaf after removing stem first were almost the same as the results with stem kept, which can also be proved from our current results that the major mis-segmentation was caused by small leaves not stem (Fig. 7). Moreover, as shown in Figs. 6 and 7, we found that the stem in Fig. 7 was almost the same as stem in Fig. 6. This suggested that the segmentation accuracy could be satisfied if the classification accuracy was satisfied even if we did not separate stem first, which might be caused by that features for classification were merged into the segmentation task.
2) Using joint-learned features to complete instance segmentation of stem and leaf was inspired by previous studies that features extracted by multitask training can improve the accuracy of each task. The benefits of the joint training features have also been proved by recent studies in computer sciences field [64], [65]

C. Comparison With Other Methods

As shown in Fig. 9, the VCNN method performed better than all the heuristic methods (MNVG, K-Mean, and DBSCAN) in both classification and segmentation tasks. The DBSCAN is an unsupervised method, which might be why it performs slightly worse than the K-means method with the supervised K class parameter. The MNVG method is designed for the stem-leaf segmentation [17] and can achieve better results with adjustable R and n parameters compared to traditional clustering methods (i.e., K-means and DBSCAN). To achieve the best results, a tradeoff between the two adjustable parameters (i.e., R and n) in the MNVG algorithm is needed. However, it is hard to achieve a good result with only one adjustable parameter (R) and one fixed parameter n, as designed in this study. By contrast, the search radius (δ_d) parameter was well trained for the VCNN, which is why VCNN achieved a higher accuracy with only one adjustable parameter.

Besides, the VCNN method was on par with the state-of-the-art deep learning methods (PointNet and PointNet++) in classification, which suggests all three networks can be used to do stem-leaf classification if they were trained with well-labeled data. However, in terms of F-score for segmentation, the VCNN showed obvious superiority to the other two CNNs. The reasons may include: first, VCNN was designed as a multitask network, which was trained to learn features that were not only for classification but also for segmentation. Therefore, VCNN can achieve high segmentation accuracy besides classification accuracy. Second, features from the PointNet and PointNet++ were not adjusted for segmentation. As shown in Fig. 12, the point distribution in segmentation feature space of PointNet++ was uniform and showed no obvious separation boundary. Therefore, the instance segmentation results (F-score) of PointNet++ were poor. Although the point distribution in segmentation feature space of PointNet

seemed to be separable, it was hard to determine the clustering radius in the clustering algorithm. By contrast, the clustering radius of our VCNN method was trained in the network, giving another reason why our method performed better than the other two methods. Finally, we think both PointNet and PointNet++ have the potential to do stem-leaf segmentation, if we modify them into a multitask neural network as with our method. In addition to the superiority in accuracy, the training time (48 h) of our method was much shorter than PointNet (110 h) and PointNet++ (114 h).

By comparing the results of the deep-learning methods (PointNet, PointNet++, and VCNN) with the heuristic methods (DBSCAN, K-means, and MNVG), we found that deep-learning methods trained by specific tasks are better than heuristic methods. For example, all classification results by deep learning methods and the segmentation results of VCNN were better than heuristic methods. These findings are consistent with previous findings that deep learning performs better when dealing with problems hard to be described by mathematical functions, because they can approximate arbitrary (smooth) functions [66], [67]. Instead, heuristic methods classify/segment objects of interests by defining hand-coded functions based on common-sense and expert knowledge. However, the stem-leaf classification and segmentation tasks are hard to be descripted by functions, which make the problem difficult to be solved by heuristic methods but suited for our VCNN. However, if deep-learning methods were not trained by specific tasks and data sets, they were not necessarily better than heuristic methods. For example, the PointNet and PointNet++ methods for segmentation in this study were inferior to the heuristic algorithms. Therefore, deep-learning methods and heuristic methods have their own strengths. Combining heuristic algorithms with deep learning may produce good results [66].

Although the VCNN achieved appreciable results in both classification and segmentation, there were still some problems, which may come from training data, the architecture of CNN, and the loss function. In the future, we can improve all these three aspects by: 1) preparing more training data containing very small leaves like the under-classified examples in Fig. 6(h) and 6(i); 2) absorbing new ideas to improve the model such as a graph convolutional network, which is designed to extract features with non-Euclidean structural data by combining heuristic ideas [68]; and 3) trying other promising loss functions such as triplet loss [69], [70] and dice loss [30], which have strengths in distinguishing details among similar components and decreasing the effect of data imbalance, respectively. In addition, the clustering algorithm from features can be substituted with the dense conditional random field method, which has been proven useful in many studies to achieve smoother results [71]–[73], and has been used as a recurrent neural network to do optimization in the SqueezeSeg network [74]. Moreover, the current method may have problems with intensive computation caused by very large numbers of empty voxels if applied to large-scale applications. This may be improved by the OctNet and Flex-Convolution methods [38], [75]–[77]. In practice, the current study has presegmented individual maize plants from the

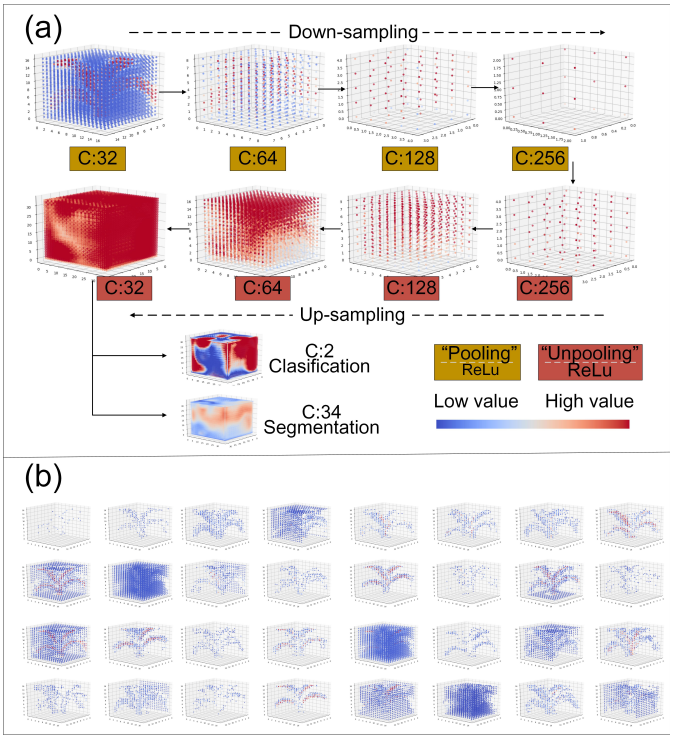


Fig. 11. (a) Representative visualization maps of extracted features at different hierarchical levels after each down sampling ("pooling"), up sampling ("unpooling"), and features for classification and segmentation. (b) All visualization maps of the extracted features after the first pooling (C:32 means the number of extracted features is 32). Other detailed visualization maps are not shown due to limited space. Colors from blue to red are stretched according to the max feature value in each voxel.

community [18] before applying the stem-leaf classification and segmentation algorithm, and the stem-leaf classification and segmentation is done at the individual plant level. Therefore, the individual maize segmentation accuracy might be a big factor influencing the stem/leaf segmentation results. Finally, our method only demonstrates the separation of the stem and leaf components of maize, although these are the major parts of maize and have strong correlations with lodging resistance, photosynthetic capacity and final yield [78], [79]. In the future, the method can be expanded to the separation of other components (e.g., tassel and fruit) if data sets are available, and be used for interdisciplinary studies [80], [81]. To promote the development of CNNs in phenotypes using LiDAR data, our data set and model will be freely accessible at https://github.com/ShichaoJin/VCNN after the study is published.

VI. Conclusion

In this study, we demonstrated the use of a voxel-based CNN to classify and segment stem and leaf instances with LiDAR scanned individual maize plants. A total of 3000 voxelized samples was used to train the VCNN, and 103 samples from three different growing stages were used to test the model performance. The VCNN method was compared with the DBSCAN, K-means, and MNVG methods. The results showed that the mean value of classification (OA) and segmentation (F) for the VCNN method were above 0.96 and

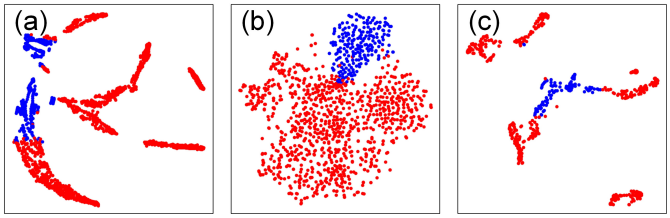


Fig. 12. Visualization map of features for stem-leaf segmentation extracted by (a) PointNet, (b) PointNet++, and (c) VCNN using t-SNE method [82]. Blue and red points: Location of stem and leaf.

around 0.90 at the instance and point level, respectively. The high accuracy mainly came from the full use of the LiDAR data and deep learning technology. The former provided good quality 3-D data, and the latter solved complex data analysis. The comparison results showed that the VCNN method performed better than traditional machine-learning-based and geometry-based methods in both classification and segmentation. When compared with state-of-the-art PointNet and PointNet++ methods, VCNN showed similar accuracy in classification and great superiority in segmentation. The high performance of VCNN may arise from the use of the discriminative loss function for segmentation, joint learning of classification and segmentation, and from fully convolutional, encode–decode, and residual learning architectures. We believe the combination of LiDAR and 3-D CNNs can greatly contribute to the separation and analysis of structural components of plants and contribute to crop phenotyping and precision agriculture in the future.

REFERENCES

[1] S. Muhammad and N. Amusa, "The important food crops and medicinal plants of north-western Nigeria," *Res. J. Agricult. Biol. Sci.*, vol. 1, no. 3, pp. 254–260, 2005.

[2] M. Minervini, H. Scharr, and S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping [applications corner]," *IEEE signal Process. Mag.*, vol. 32, no. 4, pp. 126–131, Jul. 2015.

[3] W. Li, Z. Niu, N. Huang, C. Wang, S. Gao, and C. Wu, "Airborne LiDAR technique for estimating biomass components of maize: A case study in Zhangye City, Northwest China," *Ecolog. Indicators*, vol. 57, pp. 486–496, Oct. 2015.

[4] M. Reynolds, Y. Manes, A. Izanloo, and P. Langridge, "Phenotyping approaches for physiological breeding and gene discovery in wheat," *Ann. Appl. Biol.*, vol. 155, no. 3, pp. 309–320, Dec. 2009.

[5] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 105–116.

[6] R. T. Clark *et al.*, "Three-dimensional root phenotyping with a novel imaging and software platform," *Plant Physiol.*, vol. 156, no. 2, pp. 455–465, 2011.

[7] M. Rahnemoonfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, Apr. 2017.

[8] N. Häni, P. Roy, and V. Isler, "A comparative study of fruit detection and counting methods for yield mapping in apple orchards," 2018, *arXiv:1810.09499*. [Online]. Available: https://arxiv.org/abs/1810.09499

[9] X. Xiong *et al.*, "A high-throughput stereo-imaging system for quantifying rape leaf traits during the seedling stage," *Plant Methods*, vol. 13, no. 1, p. 7, Jan. 2017.

[10] Q. Guo *et al.*, "Perspectives and prospects of LiDAR in forest ecosystem monitoring and modeling," *Chin. Sci. Bull.*, vol. 59, no. 6, pp. 459–478, 2014.

[11] M. A. Lefsky, W. B. Cohen, G. G. Parker, and D. J. Harding, "Lidar remote sensing for ecosystem studies: Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists," *BioScience*, vol. 52, no. 1, pp. 19–30, 2002.

[12] K. Zhao, J. C. Suarez, M. Garcia, T. Hu, C. Wang, and A. Londo, "Utility of multitemporal LiDAR for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux," *Remote Sens. Environ.*, vol. 204, pp. 883–897, Jan. 2018.

[13] Y. Li, Q. Guo, Y. Su, S. Tao, and K. Zhao, "Retrieving the gap fraction, element clumping index, and leaf area index of individual trees using single-scan data from a terrestrial laser scanner," *ISPRS J. Photogram. Remote Sens.*, vol. 130, pp. 308–316, Aug. 2017.

[14] S. Tao *et al.*, "A geometric method for wood-leaf separation using terrestrial and simulated Lidar data," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 10, pp. 767–776, 2015.

[15] S. Tao *et al.*, "Segmenting tree crowns from terrestrial and mobile LiDAR data by exploring ecological theories," *ISPRS J. Photogramm. Remote Sens.*, vol. 110, pp. 66–76, Dec. 2015.

[16] X. Lu, Q. Guo, W. Li, and J. Flanagan, "A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 1–12, Aug. 2014.

[17] S. Jin *et al.*, "Stem–leaf segmentation and phenotypic trait extraction of individual maize using terrestrial LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1336–1346, Mar. 2019.

[18] S. Jin *et al.*, "Deep learning: Individual maize segmentation from terrestrial Lidar data using faster R-CNN and regional growth algorithms," *Front. Plant Sci.*, vol. 9, pp. 866–875, Jun. 2018.

[19] S. Madec *et al.*, "High-Throughput phenotyping of plant height: Comparing unmanned aerial vehicles and ground LiDAR estimates," *Front. Plant Sci.*, vol. 8, p. 2002, Nov. 2017.

[20] S. Sun *et al.*, "In-field high throughput phenotyping and cotton plant growth analysis using LiDAR," *Front. Plant Sci.*, vol. 9, no. 16, p. 16, Jan. 2018.

[21] F. Hosoi and K. Omasa, "Estimation of vertical plant area density profiles in a rice canopy at different growth stages by high-resolution portable scanning lidar with a lightweight mirror," *ISPRS J. Photogram. Remote Sens.*, vol. 74, pp. 11–19, Nov. 2012.

[22] D. Ehlert, H.-J. Horn, and R. Adamek, "Measuring crop biomass density by laser triangulation," *Comput. Electron. Agricult.*, vol. 61, no. 2, pp. 117–125, 2008.

[23] J. A. Jimenez-Berni *et al.*, "High throughput determination of plant height, ground cover, and above-ground biomass in wheat with LiDAR," *Front. Plant Sci.*, vol. 9, p. 237, Feb. 2018.

[24] J. Wu, K. Cawse-Nicholson, and J. van Aardt, "3D tree reconstruction from simulated small footprint waveform lidar," *Photogramm. Eng. Remote Sens.*, vol. 79, no. 12, pp. 1147–1157, 2013.

[25] X. Yang *et al.*, "Three-dimensional forest reconstruction and structural parameter retrievals using a terrestrial full-waveform lidar instrument (Echidna)," *Remote Sens. Environ.*, vol. 135, pp. 36–51, Aug. 2013.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[27] Z. Xi, C. Hopkinson, and L. Chasmer, "Filtering stems and branches from terrestrial laser scanning point clouds using deep 3-D fully convolutional networks," *Remote Sens.*, vol. 10, no. 8, p. 1215, Aug. 2018.

[28] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," 2017, *arXiv:1711.06396*. [Online]. Available: https://arxiv.org/abs/1711.06396

[29] D. Prokhorov, "A convolutional learning system for object classification in 3-D lidar data," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 858–863, May 2010.

[30] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Stanford, CA, USA, Oct. 2016, pp. 565–571.

[31] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in Apple orchards," *J. Field Robot.*, vol. 34, no. 6, pp. 1039–1060, 2017.

[32] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, May 2016, Art. no. 3289801. [Online]. Available: https://www.hindawi.com/journals/cin/2016/3289801/

[33] A. K. Singh, B. Ganapathysubramanian, S, Sarkar, and A. Singh, "Deep Learning for plant stress phenotyping: Trends and future perspectives," *Trends Plant Sci.*, vol. 23, pp. 883–898, Oct. 2018.

[34] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar, "An explainable deep machine vision framework for plant stress phenotyping," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 8, pp. 4613–4618, 2018.

[35] S. T. Namin, M. Esmaeilzadeh, M. Najafi, T. B. Brown, and J. O. Borevitz, "Deep phenotyping: Deep learning for temporal phenotype/genotype classification," *Plant Methods*, vol. 14, pp. 66–79, Aug. 2018.

[36] X. Xiong *et al.*, "Panicle-SEG: A robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization," *Plant Methods*, vol. 13, no. 1, p. 104, Nov. 2017.

[37] D. Maturana and S. Scherer, "3D3D convolutional neural networks for landing zone detection from LiDAR," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 3471–3478.

[38] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, Jul. 2017.

[39] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, Dec. 2015, pp. 37–45.

[40] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 945–953.

[41] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," 2016, *arXiv:1612.00593*. [Online]. Available: https://arxiv.org/abs/1612.00593

[42] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*. [Online]. Available: https://arxiv.org/abs/1706.02413

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[44] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2061–2969.

[45] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015, *arXiv:1511.00561*. [Online]. Available: https://arxiv.org/abs/1511.00561

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[47] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[48] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, Jun. 2011, p. 1237.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017.

[50] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3642–3649.

[51] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[52] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, 2005.

[53] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," 2017, *arXiv:1708.02551*. [Online]. Available: https://arxiv.org/abs/1708.02551

[54] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[55] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[56] C. Goutte and E. Gaussier, *A Probabilistic Interpretation of Precision, Recall and F-Score, With Implication for Evaluation*. Berlin, Germany: Springer, 2005, pp. 345–359.

[57] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, vol. 99, Aug. 1996, pp. 226–231.

[58] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.* Oakland, CA, USA, Jun. 1967, pp. 281–297.

[59] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[60] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6016–6025.

[61] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. AAAI Conf. Artif. Inteligence*, New Orleans, LA, USA, Apr. 2018, pp. 4438–4445.

[62] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: https://arxiv.org/abs/1604.07316

[63] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: https://arxiv.org/abs/1706.05098

[64] Q.-H. Pham, D. T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields," 2019, *arXiv:1904.00699*. [Online]. Available: https://arxiv.org/abs/1904.00699

[65] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," 2019, *arXiv:1902.09852*. [Online]. Available: https://arxiv.org/abs/1902.09852

[66] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well," *J. Stat. Phys.*, vol. 168, pp. 1223–1247, Sep. 2017.

[67] L. N. Hoang and R. Guerraoui, "Deep learning works in practice. But does it work in theory," 2018, *arXiv:1801.10437*. [Online]. Available: https://arxiv.org/abs/1801.10437

[68] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: https://arxiv.org/abs/1609.02907

[69] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 815–823.

[70] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: https://arxiv.org/abs/1703.07737

[71] F. I. Alam, J. Zhou, A. W.-C. Liew, X. Jia, J. Chanussot, and Y. Gao, "Conditional Random field and deep feature learning for hyperspectral image segmentation," 2017, *arXiv:1711.04483*. [Online]. Available: https://arxiv.org/abs/1711.04483

[72] X. Liu, H. Li, W. Meng, S. Xiang, and X. Zhang, "3D point cloud classification based on discrete conditional random field," in *Proc. Int. Conf. Technol. E-Learn. Digit. Entertainment*. Cham, Switzerland: Springer, 2017, pp. 115–137.

[73] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1529–1537.

[74] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1887–1893.

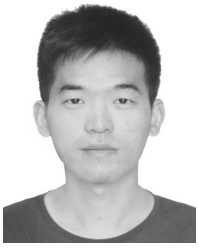
[75] F. Groh, P. Wieschollek, and H. P. A. Lensch, "Flex-convolution (million-scale point-cloud learning beyond grid-worlds)," 2018, *arXiv:1803.07289*. [Online]. Available: https://arxiv.org/abs/1803.07289

[76] G. Riegler, A. O. Ulusoys, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," 2016, *arXiv:1611.05009*. [Online]. Available: https://arxiv.org/abs/1611.05009

[77] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: Learning depth fusion from data," in *Proc. Int. Conf. 3D Vis.*, Oct. 2017, pp. 57–66.

[78] J. Xue, X. N. Qi, J. F. Shao, Z. Y. Liu, and Z. X. Li, "Effects of light intensity within the canopy on maize lodging," *Field Crops Res.*, vol. 188, pp. 133–141, Mar. 2016.

[79] Y. Wang, X. N. Qi, J. F. Shao, Z. Y. Liu, and Z. X. Li, "Effects of light intensity at full growing stage on the growth and yield of different maize varieties," *J. Jilin Agricult. Univ.*, vol. 30, no. 6, pp. 769–773, 2008.
[80] Y. Su *et al.*, "Evaluating maize phenotype dynamics under drought stress using terrestrial lidar," *Plant Methods*, vol. 15, no. 1, p. 11, Feb. 2019.
[81] Y. Jiao *et al.*, "Regulation of OsSPL14 by OsmiR156 defines ideal plant architecture in rice," *Nature Genet.*, vol. 42, pp. 541–544, May 2010.
[82] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



Shichao Jin received the B.S. degree in forestry from Huazhong Agricultural University, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Botany, Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning and LiDAR technology to solve phenotyping related challenges.

Yanjun Su received the B.E. degree in surveying and mapping engineering from the China University of Geosciences, Beijing, China, in 2009, the M.S. degree in geographic information science from the Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, in 2012, and the Ph.D. degree in environmental systems from the University of California at Merced, Merced, CA, USA, in 2017.

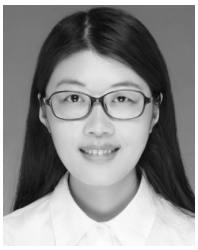
He is currently an Associate Professor with the Institute of Botany, Chinese Academy of Sciences, Beijing. His research interests include applying geographic information science and remote sensing to understand the influence of anthropogenic activities and global climate change on terrestrial ecosystems, with a particular emphasis on the terrestrial carbon cycle, terrestrial biodiversity, energy balance, and land-use/land-cover change.

Shang Gao received the B.S. degree in geography information system from Capital Normal University, Beijing, China, in 2015, and the M.S. degree with the Institute of Botany, Chinese Academy of Sciences, Beijing, in 2018.

Her research interest includes LiDAR technology and phenotyping.

Fangfang Wu received the B.S. degree in ecology from Inner Mongolia Normal University, Huhehot, China, in 2012, and the Ph.D. degree from the Institute of Botany, Chinese Academy of Sciences, Beijing, in 2018.

She currently holds a post-doctoral position with the Institute of Botany, Chinese Academy of Sciences. Her research interest includes LiDAR-based crop high-throughput phenotyping.



Qin Ma received the B.S. degree in forestry from Huazhong Agricultural University, Wuhan, China, in 2017. She is currently pursuing the Ph.D. degree with the Institute of Botany, Chinese Academy of Sciences, Beijing, China.

Her research interest includes remote sensing techniques to discover the relationship of forest structure biodiversity and function diversity.

Kexin Xu received the B.S. degree in forestry from Northwest Agriculture and Forestry University, Yangling, China, in 2017. She is currently pursuing the M.S. degree with the Institute of Botany, Chinese Academy of Sciences, Beijing, China.

Her research interest includes LiDAR-based grassland application.

Qin Ma received the B.S. degree in geography from Nanjing University, Nanjing, China, in 2011, the M.S. degree in geography from Western University, London, ON, Canada, in 2013, and the Ph.D. degree in environmental systems from the University of California at Merced, Merced, CA, USA, in 2018. She is currently an Assistant Professor with the Department of Forestry, Mississippi State University, Starkville, MS, USA. Her research focuses on using remote sensing and spatial techniques to map, monitor, and model vegetation structural and functional changes in response to human activities and climate change.

Tianyu Hu received the B.S. degree in ecology from China Agriculture University, Beijing, China, in 2008, and the Ph.D. degree from the Institute of Botany, Chinese Academy of Sciences, Beijing, in 2014.

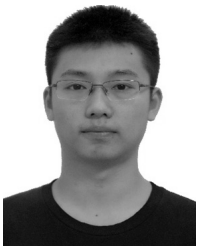
He is currently an Assistant Professor with the Institute of Botany, Chinese Academy of Sciences. His research interests include LiDAR technology and dynamic vegetation model to understand forest ecosystems, especially in forest structure, function, and biodiversity.

Jin Liu received the B.S. degree in ecology from Inner Mongolia University, Hohhot, China, in 2004, and the Ph.D. degree from the Institute of Botany, Chinese Academy of Sciences, Beijing, China, in 2012.

She is currently an Assistant Professor with the Institute of Botany, Chinese Academy of Sciences. Her research interests include the quantification of grassland BEF utilizing near-surface remote sensing technology, such as UAV LiDAR and UAV hyperspectral, to monitor and evaluate vegetation differentiation pattern under grazing at the local landscape scale.

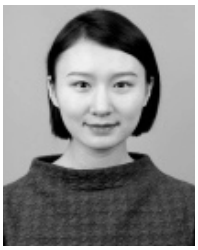
Shuxin Pang received the B.S. degree in geographic information systems (GIS) from Northwest A&F University, Yangling, China, in 2012, and the M.S. degree from the Institute of Botany, Chinese Academy of Sciences, Beijing, China, in 2016.

He is currently an Assistant Engineer with the Institute of Botany, Chinese Academy of Sciences. His research interest includes LiDAR technology and phenotyping.



Hongcan Guan received the B.S. degree in remote sensing science and technology from Wuhan University, Wuhan, China, in 2014, and the M.S. degree in electronics and communication engineering from the Institute of Opto-electronics, Chinese Academy of Sciences, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the Institute of Botany, Chinese Academy of Sciences, Beijing.

His research interest includes remote sensing technology to solve vegetation mapping related challenges.



Jing Zhang received the B.S. degree in landscape architecture from Sichuan University, Chengdu, China, in 2018. She is currently pursuing the master's degree with the Institute of Botany, Chinese Academy of Sciences, Beijing, China.

Her research interest includes exploring the application of LiDAR technology in urban planning.

Qinghua Guo received the B.S. degree in environmental science and the M.S. degree in remote sensing and geographic information systems (GIS) from Peking University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in environmental science from the University of California at Berkeley, Berkeley, CA, USA, in 2005.

He is currently a Professor with the Institute of Botany, Chinese Academy of Sciences, Beijing. He is also an Adjunct Professor and a member of the Founding Faculty, School of Engineering, University of California at Merced, Merced, CA, USA. His research interests include GIS and remote sensing algorithm development and their environmental applications, such as object-based image analysis, geographic one-class data analysis, and LiDAR data processing.