

KAT4IA: K-Means Assisted Training for Image Analysis of Field-Grown Plant Phenotypes

Xingche Guo¹, Yumou Qiu^{1, *}, Dan Nettleton¹, Cheng-Ting Yeh^{2, 3}, Zihao Zheng³, Stefan Hey³, and Patrick S. Schnable^{2, 3, †}

¹Department of Statistics, Iowa State University

²Plant Sciences Institute, Iowa State University

³Department of Agronomy, Iowa State University

*yumouqiu@iastate.edu

†schnable@iastate.edu

ABSTRACT

High-throughput phenotyping enables the efficient collection of plant trait data at scale. One example involves using imaging systems over key phases of a crop growing season. Although the resulting images provide rich data for statistical analyses of plant phenotypes, image processing for trait extraction is required as a prerequisite. Current methods for trait extraction are mainly based on supervised learning with human labelled data or semi-supervised learning with a mixture of human labelled data and unsupervised data. Unfortunately, preparing a sufficiently large training data is both time and labor intensive. We describe a self-supervised pipeline (KAT4IA) that uses K-means clustering on greenhouse images to construct training data for extracting and analyzing plant traits from an image-based field phenotyping system. The KAT4IA pipeline includes these main steps: self-supervised training set construction, plant segmentation from images of field-grown plants, automatic separation of target plants, calculation of plant traits, and functional curve fitting of the extracted traits. To deal with the challenge of separating target plants from noisy backgrounds in field images, we describe a novel approach using row-cuts and column-cuts on images segmented by transform domain neural network learning, which utilizes plant pixels identified from greenhouse images to train a segmentation model for field images. This approach is efficient and does not require human intervention. Our results show that KAT4IA is able to accurately extract plant pixels and estimate plant heights.

1 Introduction

2 One type of high-throughput phenotyping involves taking images of hundreds to thousands of plants simultaneously and
3 continuously throughout their growth period. Substantial advancements have been made by engineers and plant scientists to
4 enable large-scale collection of plant images and sensor data in greenhouses and fields [Chéné et al. \(2012\)](#); [Araus and Cairns](#)
5 [\(2014\)](#); [Hairmansis et al. \(2014\)](#); [Fahlgren et al. \(2015\)](#); [Lin \(2015\)](#); [McCormick et al. \(2016\)](#); [Xiong et al. \(2017\)](#). Figure 1

6 shows an example implemented by the Plant Science Institution (PSI) at Iowa State University, where cameras are placed in
7 front of each row of plants in a field. These cameras are designed to take side-view photos every 15 minutes from 8am to
8 5pm each day. Side-view images provide access to different plant traits as compared to top-down images generated by gantry
9 systems and UAVs (unmanned aerial vehicles). From the resulting images, we are able to process and extract phenotypic
10 features such as plant height, width and size, and use those extracted features for subsequent genetic analyses. As compared to
11 cameras mounted on mobile ground-based robots, using a system of this type allows simultaneous imaging of all plants, which
12 offers advantages in understanding genetic variation in plant responses to varying environmental conditions.

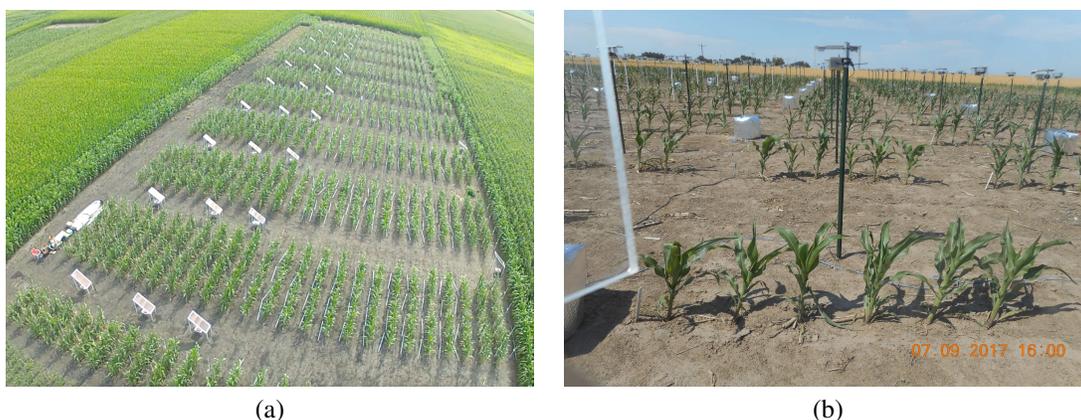


Figure 1. Left panel: an overview photo of the Iowa State field phenotyping system; right panel: raw RGB images of maize plants captured from the phenotyping facility.

13 Because high-throughput systems of this type can generate many images per day, image processing is generally required
14 to extract numerical measurements of plant traits for downstream analyses (Hartmann et al., 2011; Li et al., 2014; Araus and
15 Cairns, 2014; Choudhury et al., 2018; Adams et al., 2020). Plant object segmentation is the fundamental step in extracting
16 phenotypic features from images (Hamuda et al., 2016; Ge et al., 2016). There are existing data analysis tools built for specific
17 phenotyping systems, for example, Field Scanalyzer (Virlet et al., 2017) by LemnaTec and CropSight (Reynolds et al., 2019),
18 which uses Leaf-GP (Zhou et al., 2017) for image processing. Those tools are all based on thresholding for image segmentation,
19 which is accurate for greenhouse images, but less so for field images. Moreover, those tools are designed for top-view images
20 and cannot be directly applied to side-view images. Image segmentation and trait extraction are still the current bottlenecks in
21 many field phenotyping experiments. There are also systems, such as PlantEye by Phenospex, that generate and analyze 3D
22 images obtained from above. However, 3D imaging technologies are expensive. Due to constraints, it is generally not possible
23 to deploy hundreds of 3D lasers on large numbers of genotypes.

24 Separating plants from background is much easier for greenhouse images where the background is homogeneous (usually
25 white). Under such conditions a thresholding algorithm can often provide satisfactory results (Ge et al., 2016; Choudhury et al.,
26 2018). Thresholding is the simplest and the most commonly used method for image segmentation (Hartmann et al., 2011;
27 Davies, 2012). Segmentation often involves classifying pixels using a cut-off value for pixel intensities. Thresholding can be
28 applied on the average of red, green and blue channels, on the green-contrast intensity (Ge et al., 2016), or on both (Wang et al.,

29 2020).

30 However, thresholding methods do not perform well for field images, which typically have quite noisy backgrounds. As
31 an example, the background in Figure 1 is a mixture of dirt and plant materials on the ground, poles and silver heat shields
32 that cover phenotyping equipment, and plant shadows. Figure 2 illustrates the performance of a thresholding method on ISU
33 field images of maize, where a smaller thresholding value (0.04) maintains most parts of the plants but retains much of the
34 background noise, while a larger thresholding value (0.08) removes most of the background noise but misses many plant pixels.
35 Of particular concern, the ideal threshold for a given image is sensitive to the environment and time at which the image was
36 taken. Hence, tuning thresholding values requires extensive human intervention and introduces an additional source of human
37 bias.



Figure 2. Thresholding segmentation method for Figure 1 using green-contrast intensity with weights $(-\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, -\frac{1}{\sqrt{6}})$, and threshold level 0.04 (left panel) and 0.08 (right panel).

38 A well-segmented plant image is key to accurate feature extraction, but traits such as plant height and width are particularly
39 sensitive to background noise in images. To improve thresholding methods for greenhouse images, Adams et al. (2020) made a
40 thorough comparison for supervised learning methods trained on pixel intensities of plant RGB images acquired in a greenhouse,
41 where the training data were obtained by unsupervised K-means clustering Johnson et al. (2002); Klukas et al. (2014). They
42 demonstrated that neural network models are more accurate and robust at segmentation than traditional thresholding methods.
43 For field imaging systems, there have been an increasing number of applications of convolutional neural networks (CNN) to
44 plant phenotype extraction in recent years. Miao et al. (2019) considered leaf counting of maize by a relatively shallow CNN;
45 Lu et al. (2017) employed deep CNN structures to count the number of tassels on field-grown maize plants; Aich et al. (2018)
46 used CNNs for estimating emergence and biomass of wheat plants. Other applications of CNNs on field images are described
47 in Mohanty et al. (2016); Ubbens and Stavness (2017); Namin et al. (2018). U-net (Ronneberger et al., 2015), which uses
48 an auto encoder and decoder, is a recently developed popular CNN method for image segmentation. The idea of U-net is to
49 reconstruct an original image from its low-dimensional latent representation learned from the convolution of local structures of
50 the training data. Despite the satisfactory performance of U-net on feature extraction, preparing the training data and annotating
51 field images is still time and labor consuming because the field images are of high-resolution with noisy backgrounds.

52 To overcome the obstacle of preparing training data for field images, we provide the KAT4IA pipeline for plant feature
53 extraction from field phenotyping systems based on a self-supervised learning algorithm for plant segmentation. The idea of
54 self-supervised learning originates from semi-supervised learning (Zhu and Goldberg, 2009; Zhu et al., 2003; Kingma et al.,
55 2014), which is a machine learning approach that combines a small amount of labeled data with a large amount of unlabeled
56 data for training. Neural network-based semi-supervised learning approaches can be found in (Weston et al., 2012; Rasmus
57 et al., 2015). Semi-supervised learning also has applications in plant phenotyping. For example, (Ghosal et al., 2019) considered
58 a weakly supervised deep learning framework for sorghum head detection and counting, where the initial model is trained by a
59 small dataset and is used to annotate new data. The annotation is then verified by human expert raters and fed back into the
60 network to increase the size of training data. The proposed self-supervised learning approach generalizes semi-supervised
61 learning methods in the sense that no human labelled data are needed in the proposed approach. Self-supervised learning means
62 our KAT4IA algorithm prepares the training data for in-field plant segmentation by itself without human labelling. This is
63 possible for our problem because pixel intensities of greenhouse plants are similar to those of in-field plants, and greenhouse
64 plant pixels can be easily obtained by unsupervised learning methods, like the K-means clustering algorithm. KAT4IA is
65 able to automatically and robustly calculate plant traits from the ISU phenotyping system as shown in Figure 1, and to fit a
66 non-decreasing functional curve for the extracted traits over the plant growth period. Compared to the method of Adams et al.
67 (2020) for greenhouse images, our pipeline has the following innovations: (i) extends the plant segmentation method to field
68 images by transform domain learning; (ii) builds an automatic pipeline to separate the target plants and measure their traits; (iii)
69 uses a non-parametric monotone fitting of plant traits that is free of model assumptions.

70 An important step in KAT4IA is to obtain an accurate segmentation of plants from field images. We construct a transform
71 domain self-supervised neural network model, which uses plant pixels obtained by K-means clustering of pixels in greenhouse
72 images, along with background pixels from field images to train segmentation models. This self-supervised method, which
73 is novel in plant phenotypic analysis, can automatically and efficiently generate a large amount of supervised data by using
74 plant pixels from greenhouse images and background pixels from field images as the training pixels. It is easy to implement
75 and avoids expensive manual labelling for preparing training data. Post-processing (Vibhute and Bodhe, 2012; Davies, 2012;
76 Hamuda et al., 2016; Gehan et al., 2017) of the segmented image from the neural network model can be applied, such as
77 median blur, erosion and dilation operations. Using the segmented images, row-cut and column-cut algorithms in the pipeline
78 were developed to separate the target plants by identifying the peaks of plant pixel proportions in image rows and columns.
79 Plant features are then measured for each separated plant based on the segmented image. We also describe a refined feature
80 extraction algorithm by pooling information of plant locations from a sequence of images taken over time in the same row
81 of an experiment. In the last step, we fit a non-parametric and non-decreasing functional curve for the extracted plant trait.
82 The advantages of non-parametric functional fitting over parametric modeling and point-wise analysis of variance for plant
83 growth dynamics are discussed in Xu et al. (2018). Our method restricts the fitted curve to be non-decreasing which leads to a
84 more accurate estimation for growth curve than the approach of Xu et al. (2018). Although we mainly focus on plant height

85 measurement in this paper, our procedure can be easily extended to extract other plant traits such as size and width.

86 **The KAT4IA Method**

87 The primary interest of this paper is to automatically extract heights of all foreground plants in images recorded by cameras in
88 the field (see Figure 1), and to use the heights obtained from sequences of photos to estimate plant growth curves. The work
89 flow from the original RGB images to the fitted growth curve for each plant is summarized in Figure 3. The main steps are
90 enumerated as follows. Detailed procedures for each step are explained in the subsequent subsections.

- 91 1. Construct the training data set for plant and background pixels, whereby the plant pixels are obtained using the K-means
92 clustering algorithm applied on plant images from a greenhouse.
- 93 2. Perform image segmentation using a neural network that classifies each pixel into 0 or 1 based on the RGB intensities of
94 the training data, where 0 denotes background and 1 denotes plant.
- 95 3. Identify plants of interest and measure their heights from the segmented images.
- 96 4. Calculate the heights of plants from a sequence of images over the growing season.
- 97 5. Estimate a plant growth curve using non-parametric regression with a non-decreasing mean function for each plant.

98 **Image data**

99 The image data used in this paper were taken from a rainfed (i.e., non-irrigated) field near Grant, Nebraska in 2017. One camera
100 was installed for each row in two replications of 103 and 101 genotypes, respectively. Each row in each replication included up
101 to six plants of a single genotype. Photos were taken at a frequency of 15 minutes, and the average number of photos taken by
102 each camera was 1,719 and 1,650 respectively for the two replications. We applied KAT4IA pipeline to estimate growth curves
103 for all the plant photos taken from the two replications. The raw field photos are high resolution (5152×3864) RGB images
104 with intensity values of red, green, and blue channels between 0 and 255 for each pixel. We normalized the pixel intensities by
105 dividing by 255, producing floating point numbers between 0 and 1. To increase computation efficiency, we also re-scaled the
106 image resolution to 1000×750 .

107 **Self-supervised learning**

108 We considered self-supervised learning to classify each pixel of a field image into either a plant class or a background class.
109 As preparing accurate training data is the most labor intensive and time consuming step in supervised learning, we deployed
110 an efficient self-supervised learning method to automatically construct training data with labeled pixels for field images. To
111 prepare training data for the background, it is straightforward to crop the image into pieces that only include the background.
112 All the pixels in those pieces of images are labeled as background. For example, see the second panel in Figure 3, where the

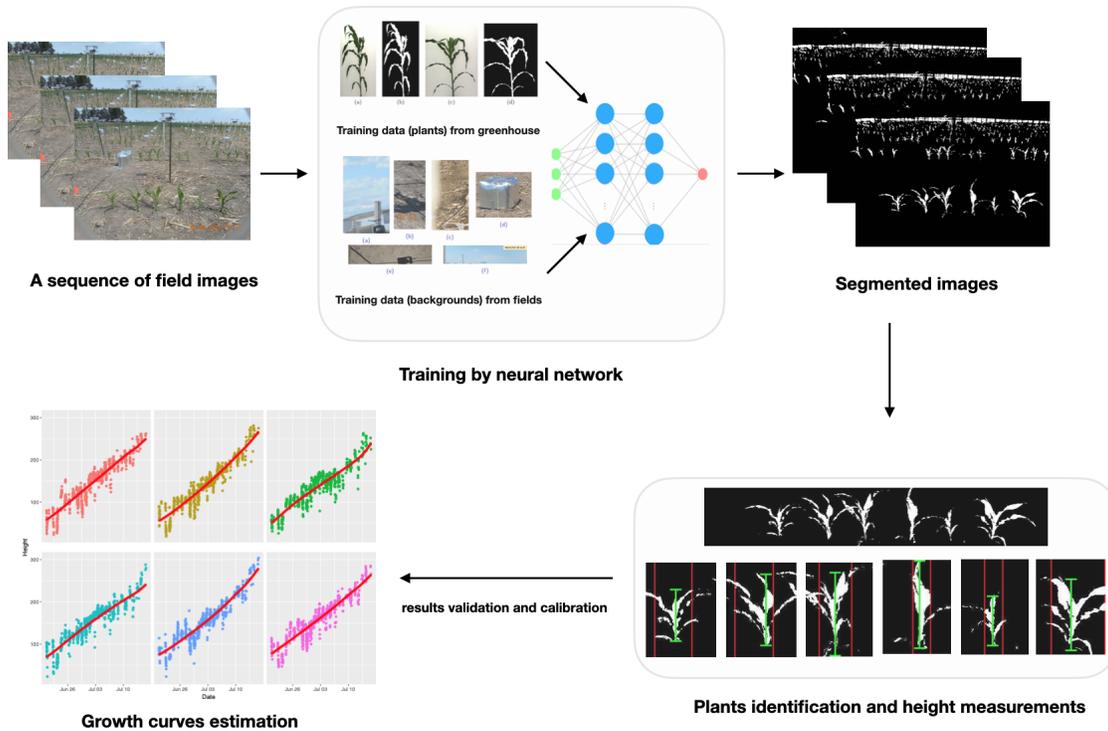


Figure 3. KAT4IA diagram. Subfigures from top left clockwise to bottom left illustrate the algorithm workflow from the original RGB images to the fitted growth curves.

113 crops of background images include the dirt and plant material on the ground, sky, shadows, and the phenotyping equipment
 114 (e.g., the poles and silver heat shields).

115 To obtain training data for the the plant class, however, it would be time-consuming to accurately crop the plant parts
 116 because of their irregular shapes and the noisy backgrounds in field images. Instead, we used plant pixels obtained from
 117 greenhouse images to train a model for field images. Specifically, we used images of plants that had been photographed in a
 118 well-controlled imaging chamber, where the backgrounds are much less noisy than field images. By cropping the greenhouse
 119 images, we obtained part of the plant in front of a background with a universal color; see panel (a) in Figure 4 as an example.
 120 This can be easily accomplished for greenhouse images. Because the cropped greenhouse images have only two distinct classes,
 121 the K-means clustering algorithm using a Euclidean distance metric can easily separate the plant pixels from the background
 122 pixels; see panel (b) in Figure 4 as the clustering result from the original image in panel (a). All the extracted plant pixels
 123 from K-means algorithm were collected as training samples of the plant class for field images. From panel (c) in Figure 4, we
 124 know that K-means clustering should not be applied on field images as it only works well for plant images with a universal
 125 background (Adams et al., 2020).

126 The key idea is to use the pixels from greenhouse plant images to train the pixel identifier for field images. Kernel
 127 density estimates of green contrast intensities for field background pixels, field-grown plant pixels and greenhouse plant pixels
 128 are shown in Figure S1 in the supplementary material. From the figure, we see that although the green contrast density of

129 greenhouse pixels is different from that of field-grown plant pixels, both densities deviate substantially from the distribution
130 for field background pixels. The green contrast intensities for field-grown plant pixels tend to be much closer to the green
131 contrast intensity distribution for greenhouse plant pixels than to the distribution for field background pixels. Thus, a classifier
132 built on the greenhouse plant pixels and field background pixels is able to separate the field-grown plants from background.
133 Despite the changing lighting conditions in the field, our learning method produced good segmentation results under various
134 field conditions and at different times of day, as demonstrated in the results section and the supplementary material section S4.
135 Note that there is no need to have a perfect segmentation of the whole plant from the greenhouse, as we only need part of the
136 plant pixels where separation from the background is easy and can be done by K-means clustering. Both the procedures to
137 construct training data for the background and plant classes are easy to implement without human labeling and annotation. This
138 makes supervised learning for plant segmentation possible at the pixel level.

139 Compared to traditional image segmentation like thresholding, our proposed method yields a more accurate results as
140 indicated by Figure S2 in the supplementary material. Our proposed method is very efficient because we do not need the
141 time-consuming and labor-expensive process of human labelling.

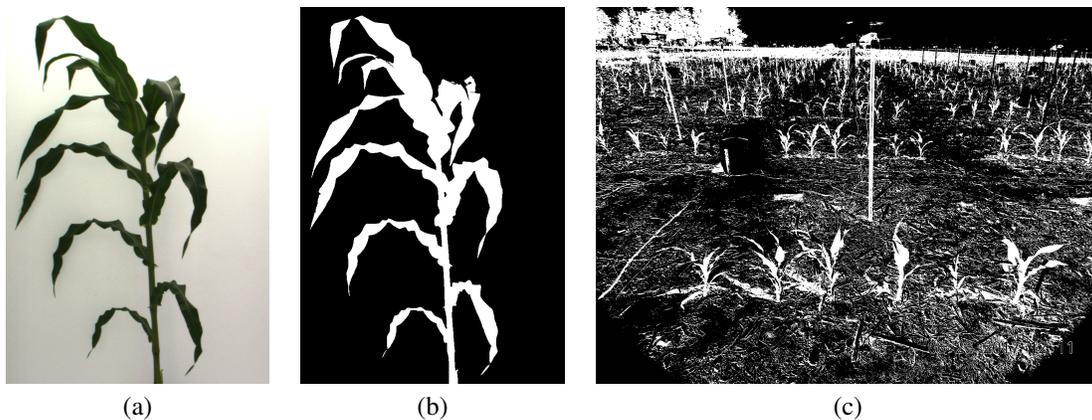


Figure 4. An example of training data (plant class) acquisition. Panel (a) is a cropped greenhouse images; panel (b) is the clustering result using the K-means algorithm ($K = 3$). The white parts are used subsequently as training data for the plant class. The number of clusters K could be chosen as 2. For $K = 3$, the third class gives the edge of the plant. Panel (c) presents the results of the K-means algorithm directly applied on a field image, which can not separate the plant pixels.

142 Segmentation by neural network

143 We used a training dataset generated as described above that consisted of 598,219 plant pixels from 6 greenhouse images and
144 2,728,415 background pixels in 19 cropped snippets from 6 field images of different environment conditions. For each pixel,
145 we used its RGB intensities and those of the surrounding eight pixels (i.e., 3×3 pixels) as the input features. This results
146 in 27 features for each pixel. Compared to neural networks with the target pixel only (i.e., no neighborhood), including the
147 neighborhood information leads to a result with less background noise. The intuition is that plant and background pixels are
148 more likely to be surrounded by pixels from their own category. In fact, the performance of neural networks with the target pixel
149 only is more similar to the thresholding segmentation method shown in Figure 2. Compared to neural networks using 5×5

150 neighborhood pixels as input features, our 3×3 neural network has a similar segmentation performance and lower computation
151 complexity. A more detailed comparison of neural networks with different neighbor sizes can be found in the supplementary
152 material section S2.

153 A three-layer neural network under the API Keras in R was used to train the model. Specifically, the input layer had 27
154 nodes, and the first and second hidden layers had 1,024 and 512 neurons respectively. The ReLU activation function was used
155 between the input layer and the first hidden layer as well as between the first and second hidden layers. The output layer had
156 one neuron which gives the predicted probability of a particular pixel belonging to the plant class. The sigmoid activation
157 function is used between the second hidden layer and the output layer. The dropout rates at each hidden layer were chosen to
158 be 0.45 and 0.35, respectively. The binary cross-entropy loss function with the Adam optimization algorithm (learning rate
159 = 0.001) was used to evaluate the network. Finally, we used 20 epochs with batch size 1,024 to train the model. 1% of the
160 training data were held out as a validation set before training.

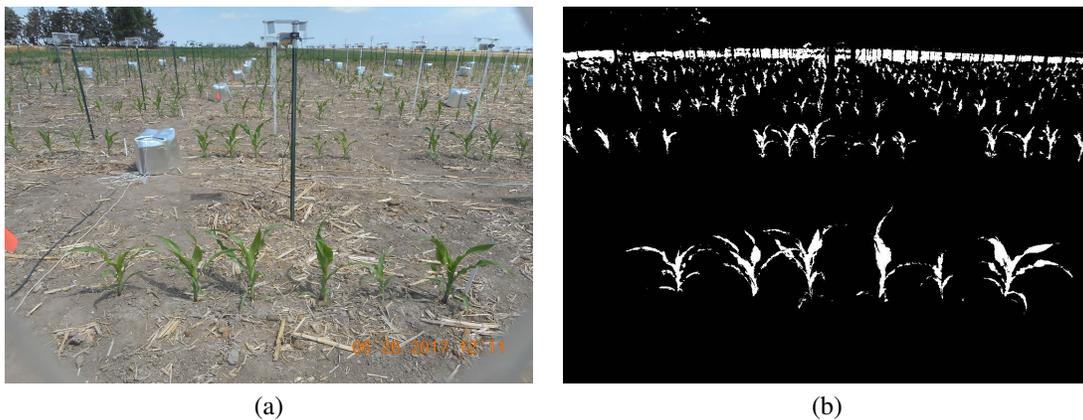


Figure 5. The original image (a) and segmentation result (b) from the self-supervised neural network model.

161 A cutoff threshold of 0.5 was used to classify the plant pixels, which means a pixel is classified as plant if its output
162 probability from the neural net model is greater than 0.5. Our method is robust to this cut-off value. More discussion and
163 results under different cut-off values can be found in the supplementary material section S3. Figure 5 provides an example of
164 the segmentation result by our neural network model. Most of the plants were precisely segmented with limited background
165 noise. Even a corn field in the extreme background near the top of the image was correctly classified as plant. In contrast, the
166 trees on the horizon were, for the most part, classified as background. More segmentation results for different plants and under
167 various environmental conditions are shown in Figure S5 in the supplementary material. From those results, we can see that the
168 proposed method is stable and robust under different weather and light conditions.

169 **Plant height measurement from a single segmented image**

170 Based on the segmented images, we aimed to measure the height of the plants in the first (most forward) row of an image. As
171 an example, there are six maize plants in the first row of Figure 5. This procedure constitutes identifying the first row by a
172 row-cut algorithm and then separating each plant in the first row by a column-cutting algorithm before measuring the individual

173 height of each plant.

174 **Row-cut algorithm**

175 To separate the first row in an image, we use a row-cut algorithm which consists of local maximum calling and region
176 identification. Specifically, row means are calculated for each pixel row of the segmented image, which gives the percentage of
177 plant pixels in each row. Then a local smoother (loess function in R) is used to smooth the row means. From Figure 6, we
178 can see multiple peaks in the row mean curve, where the bottom peak corresponds to the front row of plants. To find the local
179 maximum of the bottom peak, we threshold the row means by $R_v = 10\%$ percent of their global maximum value. This results in
180 segments of row indices with values above the threshold, where two segments are considered to be separate if they are $S_r = 10$
181 pixel rows apart. The maximum of the bottom peak is the largest row mean in the first segment at the bottom of the image. See
182 the illustration in the top right panel of Figure 6, where the red point denotes the maximum of the bottom peak (colored in
183 green) identified by the procedure. Finally, to locate the region of the bottom peak, its upper and lower boundaries are chosen
184 as the first pixel rows smaller than $R_u = 7.5\%$ and $R_l = 2.5\%$ percentage of its peak maximum when moving above and below
185 from the center of the bottom peak. See the bottom two panels in Figure 6 as an illustration of this step. Our results show that
186 this procedure can accurately separate the first row of plants and that it is robust to the tuning parameters R_v , R_u , R_l and S_r for
187 all images analyzed. However, the appropriate values of those hyper-parameters may vary in different experimental settings.

188 **Column-cut algorithm**

189 Once the targeted row of plants is obtained, we separate each plant in that row using a column-cut algorithm. This algorithm
190 is illustrated in Figure 7. Similar to the row-cut algorithm, the first step is to compute the pixel column mean values, which
191 gives the column-wise percentage of segmented plant pixels. We applied a quadratic power transformation (i.e. $f(x) = x^2$)
192 to the column means, which magnifies the column peak maximal values so that it is easier to separate different peaks, as
193 illustrated in the third step in Figure 7. Following the same strategy as the row-cut algorithm, we find the maximum for each
194 peak by thresholding the squared column means at $C_h = 20\%$ percent of the overall maximum, and obtaining segments defined
195 by column indices with values larger than this threshold. Then, segments that are at least $S_c = 50$ pixel columns apart are
196 considered to be from different peaks. The maximum value for each peak can be obtained as the largest squared column
197 means in each segment. The cuts between plants are calculated as the midpoints between the indices of two adjacent peak
198 maxima. Specifically, let $\{I_p^{(j)}\}_{j=1}^m$ be the indices of the column-mean peak maximum for the m plants. Let $I_c^{(j)}$, $j = 2, \dots, m$
199 be the indices of the cuts between plants. The left and right margin cuts are defined to be $I_c^{(1)} = \max\{I_p^{(1)} - D_I, 1\}$ and
200 $I_c^{(m+1)} = \min\{I_p^{(m)} + D_I, n_c\}$ respectively, where $D_I = \max_{j \in \{1, \dots, m-1\}} \lceil \frac{I_p^{(j+1)} - I_p^{(j)}}{2} \rceil$ and n_c is the total number of columns.

201 **Phenotype measurements**

202 After making the row and column cuts, we can measure phenotypic traits for each plant. In this study, we focused on height
203 measurement. The proposed procedure could, however, be easily adjusted to calculate plant width and size. For the height of
204 each separated plant, we first computed the column means, then found the maximum value and the corresponding index of that

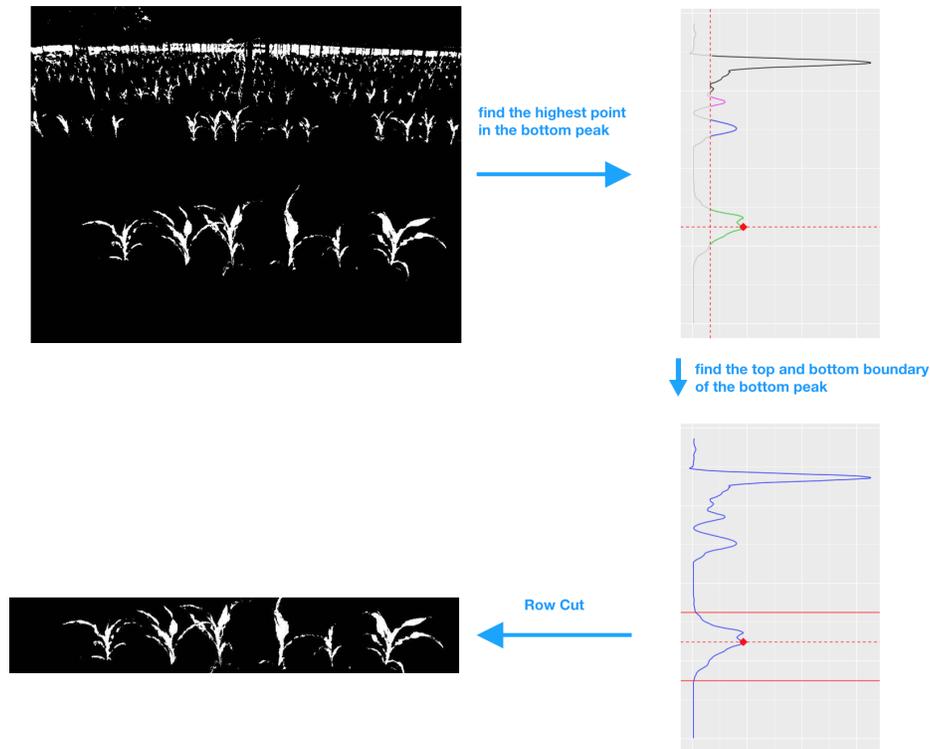


Figure 6. Diagram of the row-cut algorithm. Top left panel: the segmented image of plants from the neural network model; top right panel: the step of local maximum calling, which provides a separation of different peaks (illustrated by different colors) in the row mean curve and an identification of the maximum of the lower peak (denoted by the red point); bottom right panel: the step of peak region identification, providing the upper and lower boundaries of the bottom peak (denoted by the red solid lines); bottom left panel: the segmented and cropped first row of plants from the original image.

205 maximum. Lastly, the left and right cuts were made to retain the center part of the plant: each cut was made at the pixel column
 206 closest to the column with the highest value among columns at which less than 10% of the maximum value was reached. The
 207 row mean values for the selected center part of the plant are computed, and the plant height is calculated as the index difference
 208 between the first row from below and the first row from above with mean values larger than 2.5% of the maximal row mean
 209 value. This procedure is illustrated in Figure 8.

210 Plant height measurement for each time series of images

211 In this section, we outline a refined height measurement procedure for a sequence of plant photos taken over time by borrowing
 212 information of plant locations across the time series of images. After conducting the above procedures for image segmentation,
 213 row cut and column cuts, we can systematically study the growth trend of each separated plant over time, and refine the
 214 column-cut algorithm that is based on a single image by considering a sequence of images from the same row, as the camera
 215 positions generally remain approximately fixed throughout the experiment. Consideration of a sequence of images can help to
 216 remove problematic images and images with overlapping rows of plants from which a clear separation of the plants in the front
 217 row is difficult.

218 Figure 9 shows a set of field photos of a row of plants taken by a single camera over time. Notice that the plant locations

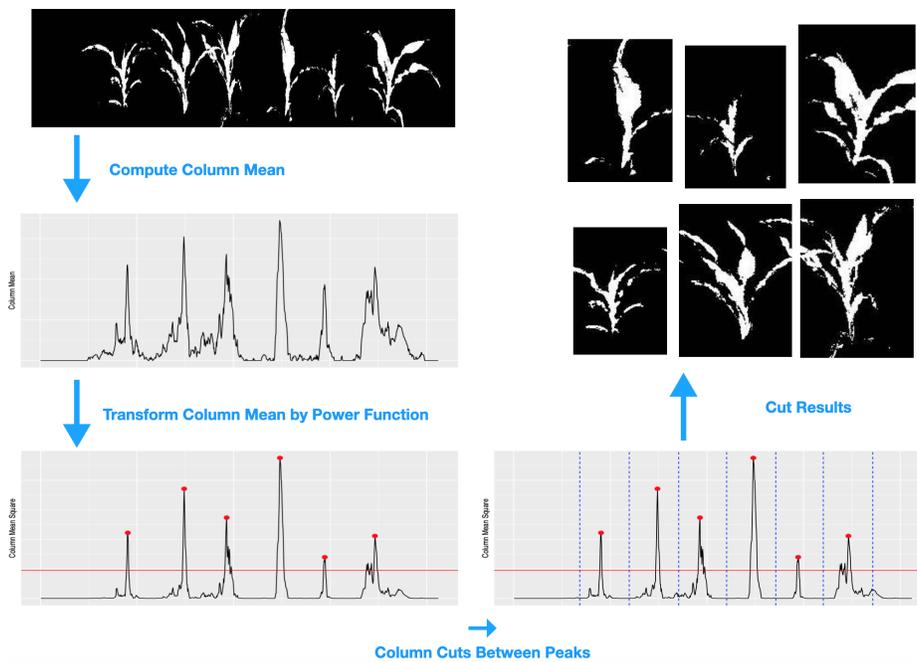


Figure 7. Diagram of the column-cut algorithm. Top left panel: the segmented first row of plants from the row-cut algorithm; middle left panel: the column mean curve; bottom left panel: the step of local maximum calling for the column mean curve, providing the maximum of each peak after the power transformation (denoted by red points); bottom right panel: the step of plant separation, where the cuts (blue dashed lines) between plants are calculated as the middle points of two adjacent peaks; top right panel: the segmented and cropped image for each plant.

219 of plants are roughly the same across different photos. However, we cannot identify all six plants from every photo due to
 220 technical issues of the camera (panels a and b where the rightmost plant is obscured), strong wind (panel e where the second
 221 and third plants overlap) or the death of particular plants. Meanwhile, the row-cut algorithm requires a separation between the
 222 first (front) row and the second (background) row of plants, so that the bottom peak of the row means are separable from other
 223 peaks; see Figure 6. When the plants in the first row overlaps with the plants in the background, as shown in panel (f) of Figure
 224 9, it is challenging to accurately measure plant height using computer vision methods. Our neural network algorithm is not able
 225 to separate the first row from the rest of the rows if they are overlapping in the perspective of the image. Hence, the current
 226 method is suitable for the earlier growth stages of field-grown plants. We explore potential solutions to this problem in the
 227 discussion.

228 To deal with the aforementioned challenges of the dynamic photos of plant growth, we have developed an algorithm to
 229 check image qualities to obtain more reliable estimates of plant height. This algorithm includes four steps as follows. Firstly,
 230 the neural network segmentation model and the row-cut algorithm are applied to every photo in the sequence, and the heights of
 231 the segmented first row from each image are computed. We apply change point detection methods (via *changept* R package)
 232 to identify jumps in the heights of the segmented rows from the sequence of images. As illustrated in the top left panel of Figure
 233 10, there is a clear jump in the row heights around July 21. This change point, denoted by the red vertical line, corresponds to
 234 the date when the front line of plants begins to overlap with the plants in the background, becoming inseparable. The current

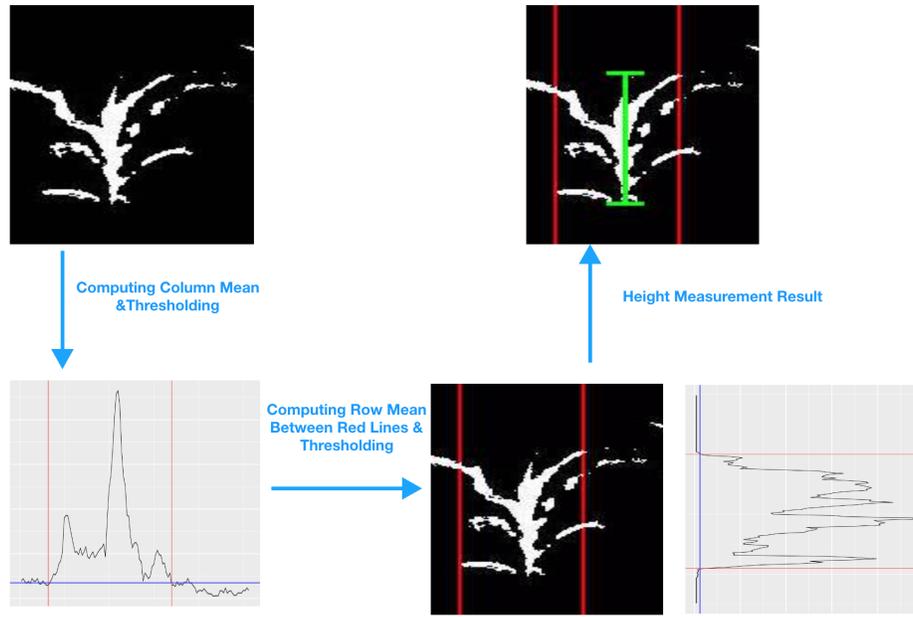


Figure 8. Diagram of the height measurement algorithm. Top left panel: the segmented image for a single plant from the row-cut and column-cut algorithms; bottom left panel: extracting the center part of the plant by thresholding (blue line) the column mean curve of the segmented image in the top left panel and identifying the left and right cuts (red lines); bottom right panel: the extracted center part (marked by two solid red lines) of the segmented image, and the height measurement by thresholding (blue line) the row mean curve of the center part of the segmented image; top right panel: the segmented image of a plant with the annotated height.

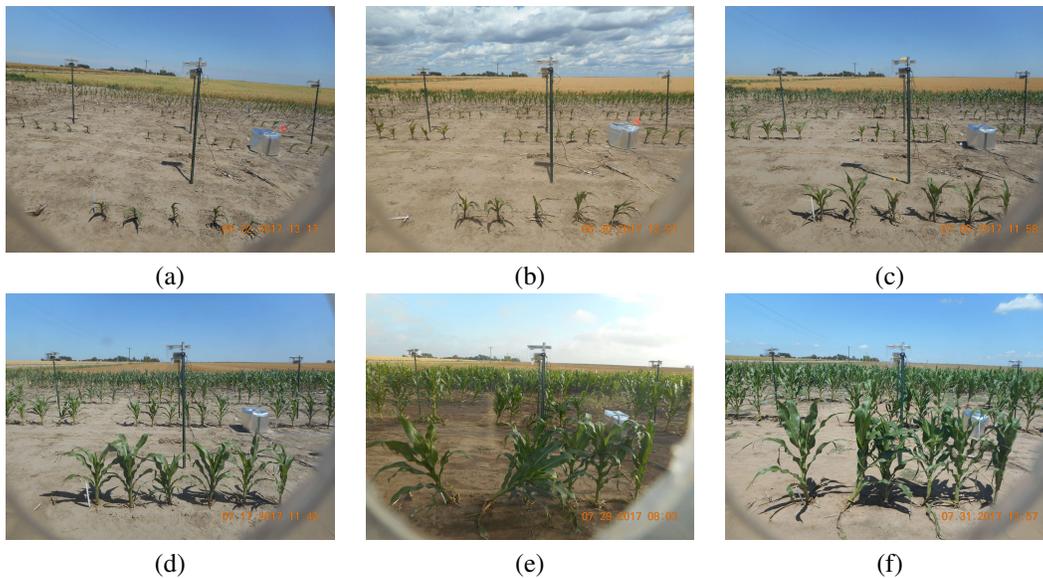


Figure 9. A sequence of field photos from a row of plants over the growth period.

235 height measurement method only works for early stages of plant growth when the target row of plants does not overlap with
 236 plants in the background. To separate plants from overlapped rows, we need to first obtain a good segmentation of all the plants
 237 that removes the background noise and then identify the targeted plants from the segmented image. The proposed method
 238 provides a solution to the first step of this process. We describe how to separate targeted plants when the rows are overlapping

239 in the discussion section. We focus on measuring the plant heights of the front row prior to this change point. Secondly, the
 240 column cuts algorithm is implemented to count the number of plants in the front row for the segmented images from step one.
 241 The mode of these counts, denoted by m , is used as an estimate for the true number of plants in a given row over time. Because
 242 six seeds are planted in each row in this experiment, the modes for most of the rows are six during the growing season. We only
 243 consider those images with the number of plants in the first row equal to its mode m . This is illustrated in the top right and
 244 bottom left panels of Figure 10, where $m = 6$ and the red points are the images with 6 identified plants over the time course. We
 245 compute the plant heights for those selected images for the time sequence of photos in the following steps.

246 Given a row (camera), let n be the number of the selected images with m identified plants from the first two steps.
 247 In the third step, we refine the column cuts for each plant in a row by pooling information of plant locations from those
 248 selected n images. Let $I_p^{(i,j)}$ be the column peak index for the j th plant in the i th photo. The average column peak index for
 249 the j th plant can be computed as $\bar{I}_p^{(j)} = n^{-1} \sum_{i=1}^n I_p^{(i,j)}$. Note that the camera might slightly shift horizontally due to wind,
 250 which affects the position of the column peaks over time in a given row. However, the distance between two adjacent peaks
 251 should remain constant. Therefore, it is reasonable to stabilize the column peak index for the j th plant in the i th photo as
 252 $\hat{I}_p^{(i,j)} = \bar{I}_p^{(j)} + \text{median}_j(I_p^{(i,j)}) - \text{median}_j(\bar{I}_p^{(j)})$, where the term $\text{median}_j(I_p^{(i,j)}) - \text{median}_j(\bar{I}_p^{(j)})$ adjusts the horizontal shift of the
 253 camera. The separation for each plant can be made at the average index of two adjacent peaks, as discussed in the ‘‘Column-cut
 254 algorithm’’ section. The red solid lines and blue dashed lines in the bottom right panel of Figure 10 show the stabilized column
 255 peaks and column cuts, respectively. Finally, we calculate the height of each separated plant as discussed in the previous section.
 256 The measured heights for the six plants in Figure 10 are shown in Figure 11.

257 Estimating growth curves

258 Plant heights are not expected to decrease during the growing season. Using the extracted heights from the plant images, we
 259 can fit a growth curve for each plant by nonparametric regression (Wahba, 1990; Fan and Gijbels, 1996). However, the classical
 260 nonparametric curve fitting methods cannot ensure the non-decreasing property for the growth curve. To fit a non-decreasing
 261 function for the plant growth, following Dette et al. (2006), we first apply a kernel-based estimation to fit an unconstrained
 262 growth curve $\hat{\mu}(t)$. Then, we construct a density estimate using the estimated values $\hat{\mu}(i/N)$ for $i = 1, \dots, N$, where N is the
 263 total number of observations over time. It can be shown that integrating the density estimate from $-\infty$ to t gives a consistent and
 264 non-decreasing estimator for $\mu^{-1}(t)$ if $\mu(t)$ is a non-decreasing function. Thus, the estimator for $\mu(t)$ is also a non-decreasing
 265 function. To make estimation more robust, outlying height measurements are detected based on the interquartile range of
 266 the residuals. Height measurements whose residuals are outside 3 times the interquartile range are ignored when fitting the
 267 non-decreasing growth curves a second time. The curves in Figure 11 are the fitted non-decreasing growth curves based on this
 268 method for six plants in one camera before the front row and the background rows overlap. Our method fit the data well with
 269 high R-square values. The goodness-of-fit results of the proposed method are reported in the supplementary material section S5.

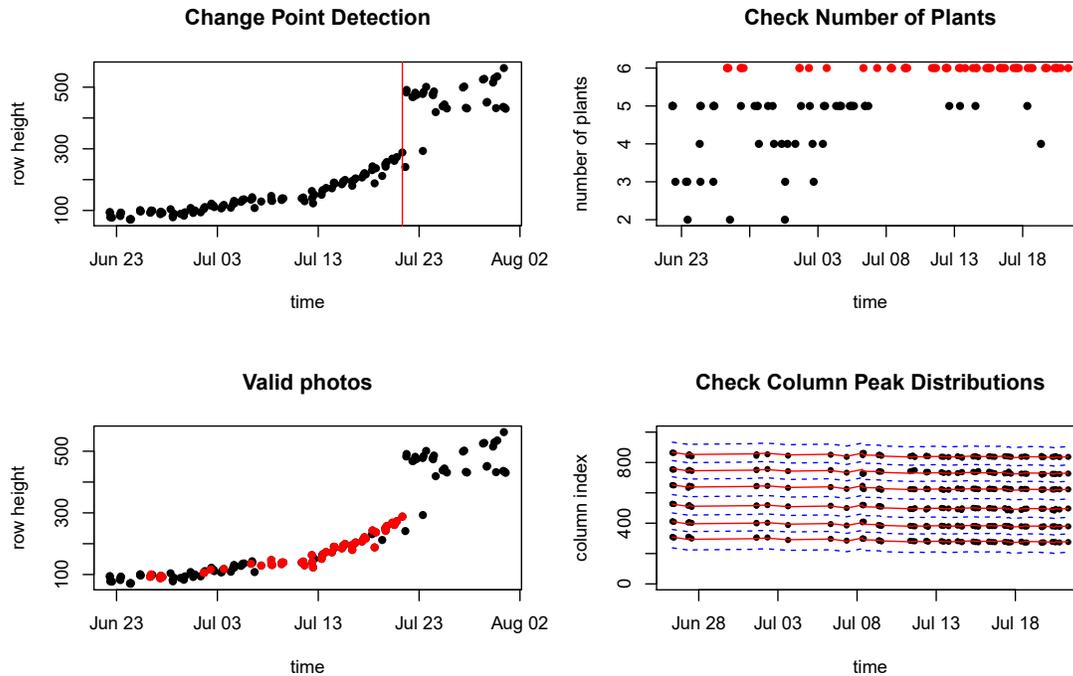


Figure 10. Refined height measurements for an exemplary sequence of images from one row. Top left panel: change point detection to identify the jump in the heights of the segmented rows, where the plants in the first row overlap with the background rows; top right panel: the number of identified plants in a given row over time; bottom left: the selected images (marked as red) for the growth curve analysis, which have 6 identified plants before row overlapping; bottom right: refining the column cuts for each image by pooling information of plant locations from other images in the same row over the growth period. The red solid lines are the estimated center of each plant over time, and the blue dashed lines are the refined column cuts.

Discussion

This paper describes a self-supervised method (K-means assisted training) to separate plants from background for field images and a computation pipeline to extract plant features (traits) from the segmented images. Our self-supervised learning approach is advantageous for high-throughput phenotypic analyses as no human-labelling is required to construct supervisory training data. The absence of tedious human labelling makes up-scaling efficient and feasible. Our KAT4IA method is easy to implement and can be broadened to provide a variety of plant phenotypic analyses. Although this paper focuses on extracting height measurements, other features can also be extracted from the segmented images. For example, topological skeletonization can be applied to the post-segmentation binary images, and leaves can be separated based on skeleton-based computer vision methods.

The idea of transforming learning that uses greenhouse images to learn field images can be applied to various feature extraction problems. As many plant features, including height and number of leaves, have been extracted from greenhouse plant images (Miao et al., 2019), we can generate pseudo-field images based on greenhouse images with their extracted plant features, and build machine learning models on those pseudo-field images to measure plant traits from field phenotyping projects.

As shown in Figure 10, the proposed method works for early stages of plant growth, during which the first row in the images does not overlap with plants in the background. Self-supervised learning methods can also be developed to separate the first

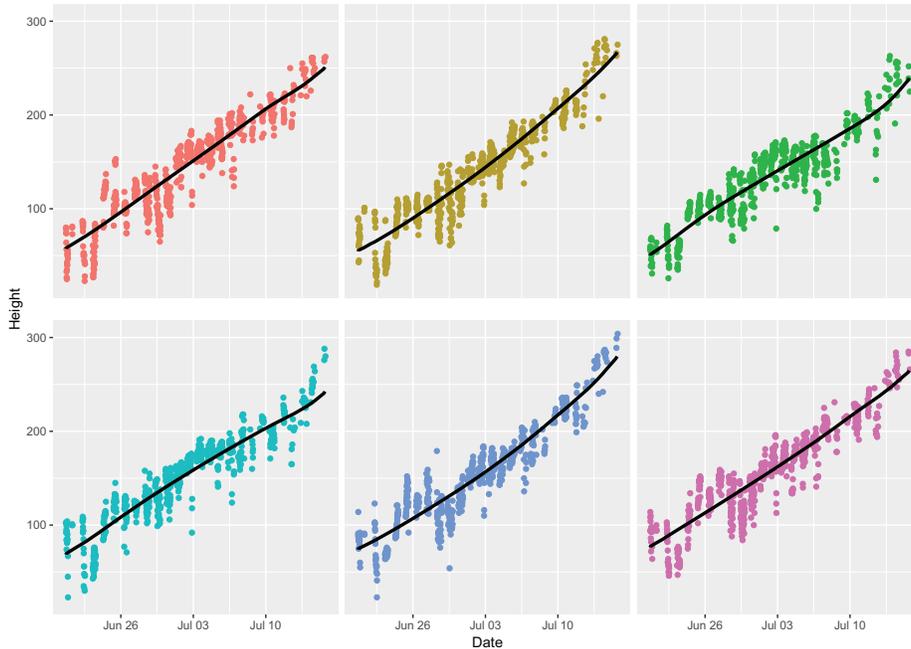


Figure 11. The fitted growth curves for each plant in a set of images from one camera. The points are the extracted plant heights from images, and the non-decreasing curves are the fitted values from the KAT4IA pipeline.

284 row from the background plants if they overlap. This can be achieved in a two-step procedure. In the first step, the proposed
 285 segmentation method would be applied to segment all plants from the background. Training data of plant pixels from the first
 286 row and the background rows can be automatically formed from the images where the first row is separable. In the second step,
 287 using the training data, a convolutional neural network model can be constructed based on the pixel intensities from a small
 288 neighborhood of each pixel. In the same way we have used greenhouse images to train self-supervised learning for field-grown
 289 plants, we can use plant images in early growth stages to form self-supervisory information for separation of plants in late
 290 growth stages.

291 The functional curve smoothing method is applied on each individual plant over time. Functional data analysis for genotype
 292 and treatment effects on plant growth can be conducted based on the fitted values from the non-decreasing functional curve.
 293 The “*implant*” package (Wang et al., 2020) can be applied on the smoothed plant traits for this purpose.

294 Currently, we do not have high-throughput field images with labeled plant pixels. In future work, results generated from
 295 our KAT4IA approach could be compared to results obtained by more labor-intensive approaches, such as using manually
 296 segmented images for supervised learning, obtaining manually measured heights of plants from images, or manually measuring
 297 plant heights in the field.

298 Finally, weeds were well controlled in our experiment, which can be seen from the original images. So, the proposed
 299 segmentation model does not consider weeds as the background. When weeds are prevalent, we could crop the part of the
 300 in-field images with weeds and use their pixels as part of the training data for the background class. A larger neighborhood size
 301 might be needed, as those surrounding pixels may be able to distinguish the structure differences between maize plants and

302 weeds.

303 **References**

- 304 Adams, Jason, Yumou Qiu, Yuhang Xu, and James C Schnable (2020), “Plant segmentation by supervised machine learning
305 methods.” *The Plant Phenome Journal*, 3, e20001.
- 306 Aich, Shubhra, Anique Josuttis, Ilya Ovsyannikov, Keegan Strueby, Imran Ahmed, Hema Sudhakar Duddu, Curtis Pozniak,
307 Steve Shirliffe, and Ian Stavness (2018), “Deepwheat: Estimating phenotypic traits from crop images with deep learning.”
308 In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 323–332, IEEE.
- 309 Araus, José Luis and Jill E Cairns (2014), “Field high-throughput phenotyping: the new crop breeding frontier.” *Trends in plant
310 science*, 19, 52–61.
- 311 Chéné, Yann, David Rousseau, Philippe Lucidarme, Jessica Bertheloot, Valérie Caffier, Philippe Morel, Étienne Belin, and
312 François Chapeau-Blondeau (2012), “On the use of depth camera for 3d phenotyping of entire plants.” *Computers and
313 Electronics in Agriculture*, 82, 122–127.
- 314 Choudhury, Sruti Das, Srinidhi Bashyam, Yumou Qiu, Ashok Samal, and Tala Awada (2018), “Holistic and component plant
315 phenotyping using temporal image sequence.” *Plant methods*, 14, 35.
- 316 Davies, E Roy (2012), *Computer and machine vision: theory, algorithms, practicalities*. Academic Press.
- 317 Dette, Holger, Natalie Neumeyer, Kay F Pilz, et al. (2006), “A simple nonparametric estimator of a strictly monotone regression
318 function.” *Bernoulli*, 12, 469–490.
- 319 Fahlgren, Noah, Malia A Gehan, and Ivan Baxter (2015), “Lights, camera, action: high-throughput plant phenotyping is ready
320 for a close-up.” *Current opinion in plant biology*, 24, 93–99.
- 321 Fan, Jianqing and Irene Gijbels (1996), *Local polynomial modelling and its applications: monographs on statistics and applied
322 probability 66*, volume 66. CRC Press.
- 323 Ge, Y., G. Bai, V. Stoerger, and J. C. Schnable (2016), “Temporal dynamics of maize plant growth, water use, and leaf water
324 content using automated high throughput rgb and hyperspectral imaging.” *Computers and Electronics in Agriculture*, 127,
325 625–632.
- 326 Gehan, Malia A, Noah Fahlgren, Arash Abbasi, Jeffrey C Berry, Steven T Callen, Leonardo Chavez, Andrew N Doust, Max J
327 Feldman, Kerrigan B Gilbert, John G Hodge, et al. (2017), “Plantcv v2: Image analysis software for high-throughput plant
328 phenotyping.” *PeerJ*, 5, e4088.

329 Ghosal, Sambuddha, Bangyou Zheng, Scott C Chapman, Andries B Potgieter, David R Jordan, Xuemin Wang, Asheesh K
330 Singh, Arti Singh, Masayuki Hirafuji, Seishi Ninomiya, et al. (2019), “A weakly supervised deep learning framework for
331 sorghum head detection and counting.” *Plant Phenomics*, 2019.

332 Hairmansis, Aris, Bettina Berger, Mark Tester, and Stuart John Roy (2014), “Image-based phenotyping for non-destructive
333 screening of different salinity tolerance traits in rice.” *Rice*, 7, 16.

334 Hamuda, Esmael, Martin Glavin, and Edward Jones (2016), “A survey of image processing techniques for plant extraction and
335 segmentation in the field.” *Computers and Electronics in Agriculture*, 125, 184–199.

336 Hartmann, Anja, Tobias Czauderna, Roberto Hoffmann, Nils Stein, and Falk Schreiber (2011), “Htpheno: an image analysis
337 pipeline for high-throughput plant phenotyping.” *BMC bioinformatics*, 12, 148.

338 Johnson, Richard Arnold, Dean W Wichern, et al. (2002), *Applied multivariate statistical analysis*, volume 5. Prentice Hall.

339 Kingma, Diederik P, Danilo J Rezende, Shakir Mohamed, and Max Welling (2014), “Semi-supervised learning with deep
340 generative models.” *arXiv preprint arXiv:1406.5298*.

341 Klukas, Christian, Dijun Chen, and Jean-Michel Pape (2014), “Integrated analysis platform: an open-source information system
342 for high-throughput plant phenotyping.” *Plant physiology*, 165, 506–518.

343 Li, Lei, Qin Zhang, and Danfeng Huang (2014), “A review of imaging techniques for plant phenotyping.” *Sensors*, 14,
344 20078–20111.

345 Lin, Yi (2015), “Lidar: An important tool for next-generation phenotyping technology of high potential for plant phenomics?”
346 *Computers and electronics in Agriculture*, 119, 61–73.

347 Lu, Hao, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen (2017), “Tasselnet: counting maize tassels in the wild via
348 local counts regression network.” *Plant methods*, 13, 79.

349 McCormick, Ryan F, Sandra K Truong, and John E Mullet (2016), “3d sorghum reconstructions from depth images identify qtl
350 regulating shoot architecture.” *Plant physiology*, 172, 823–834.

351 Miao, Chenyong, Thomas P Hoban, Alejandro Pages, Zheng Xu, Eric Rodene, Jordan Ubbens, Ian Stavness, Jinliang Yang, and
352 James C Schnable (2019), “Simulated plant images improve maize leaf counting accuracy.” *bioRxiv*, 706994.

353 Mohanty, Sharada P, David P Hughes, and Marcel Salathé (2016), “Using deep learning for image-based plant disease detection.”
354 *Frontiers in plant science*, 7, 1419.

355 Namin, Sarah Taghavi, Mohammad Esmaeilzadeh, Mohammad Najafi, Tim B Brown, and Justin O Borevitz (2018), “Deep
356 phenotyping: deep learning for temporal phenotype/genotype classification.” *Plant methods*, 14, 66.

357 Rasmus, Antti, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko (2015), “Semi-supervised learning with
358 ladder networks.” *arXiv preprint arXiv:1507.02672*.

359 Reynolds, Daniel, Joshua Ball, Alan Bauer, Robert Davey, Simon Griffiths, and Ji Zhou (2019), “Cropsight: a scalable and open-
360 source information management system for distributed plant phenotyping and iot-based crop management.” *Gigascience*, 8,
361 giz009.

362 Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015), “U-net: Convolutional networks for biomedical image segmenta-
363 tion.” In *International Conference on Medical image computing and computer-assisted intervention*, 234–241, Springer.

364 Ubbens, Jordan R and Ian Stavness (2017), “Deep plant phenomics: a deep learning platform for complex plant phenotyping
365 tasks.” *Frontiers in plant science*, 8, 1190.

366 Vibhute, Anup and SK Bodhe (2012), “Applications of image processing in agriculture: a survey.” *International Journal of*
367 *Computer Applications*, 52.

368 Virlet, Nicolas, Kasra Sabermanesh, Pouria Sadeghi-Tehran, and Malcolm J Hawkesford (2017), “Field scanalyzer: An
369 automated robotic field phenotyping platform for detailed crop monitoring.” *Functional Plant Biology*, 44, 143–153.

370 Wahba, Grace (1990), *Spline models for observational data*, volume 59. Siam.

371 Wang, Ronghao, Yumou Qiu, Yuzhen Zhou, Zhikai Liang, and James C Schnable (2020), “A high-throughput phenotyping
372 pipeline for image processing and functional growth curve analysis.” *Plant Phenomics*, 2020.

373 Weston, Jason, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert (2012), “Deep learning via semi-supervised embedding.”
374 In *Neural networks: Tricks of the trade*, 639–655, Springer.

375 Xiong, Xiong, Lejun Yu, Wanneng Yang, Meng Liu, Ni Jiang, Di Wu, Guoxing Chen, Lizhong Xiong, Kede Liu, and Qian Liu
376 (2017), “A high-throughput stereo-imaging system for quantifying rape leaf traits during the seedling stage.” *Plant Methods*,
377 13, 7.

378 Xu, Yuhang, Yumou Qiu, and James C Schnable (2018), “Functional modeling of plant growth dynamics.” *The Plant Phenome*
379 *Journal*, 1.

380 Zhou, Ji, Christopher Applegate, Albor Dobon Alonso, Daniel Reynolds, Simon Orford, Michal Mackiewicz, Simon Griffiths,
381 Steven Penfield, and Nick Pullen (2017), “Leaf-gp: an open and automated software application for measuring growth
382 phenotypes for arabidopsis and wheat.” *Plant methods*, 13, 1–17.

383 Zhu, Xiaojin, Zoubin Ghahramani, and John D Lafferty (2003), “Semi-supervised learning using gaussian fields and harmonic
384 functions.” In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 912–919.

385 Zhu, Xiaojin and Andrew B Goldberg (2009), “Introduction to semi-supervised learning.” *Synthesis lectures on artificial*
386 *intelligence and machine learning*, 3, 1–130.

387 **Additional information**

388 The R code of the proposed pipeline, sample image data and description are available on Github at [https://github.com/](https://github.com/xingcheg/Plant-Traits-Extraction)
389 [xingcheg/Plant-Traits-Extraction](https://github.com/xingcheg/Plant-Traits-Extraction).

390 **Conflict of Interest**

391 The authors declare that they do not have any commercial or associative interest that represents a conflict of interest in
392 connection with the work.

393 **Author contributions statement**

394 X.G., Y.Q. and D.N. developed the pipeline, conducted the analysis, and wrote the manuscript with editorial assistance by
395 P.S.S.; C.T.Y., Z.Z., S.H. and P.S.S. designed and conducted the data collection experiment; C.T.Y. managed image data storage
396 and access. All authors with the exception of S.H. participated in brain-storming sessions.

397 **Acknowledgement**

398 This research is supported in part by the US National Science Foundation under grant HDR:TRIPPODS 19-34884, the United
399 States Department of Agriculture National Institute of Food and Agriculture Hatch project IOW03617, the Office of Science
400 (BER), U.S. Department of Energy, Grant no. DE-SC0020355, and the Iowa State University Plant Sciences Institute Scholars
401 Program. The authors thank Colton McNinch for assistance with the implementation of the field experiment.

Supplementary Material for “KAT4IA: K-Means Assisted Training for Image Analysis of Field-Grown Plant Phenotypes”

Xingche Guo¹, Yumou Qiu^{1, *}, Dan Nettleton¹, Cheng-Ting Yeh^{2, 3}, Zihao Zheng³, Stefan Hey³, and Patrick S. Schnable^{2, 3, †}

¹Department of Statistics, Iowa State University

²Plant Sciences Institute, Iowa State University

³Department of Agronomy, Iowa State University

*yumouqiu@iastate.edu

†schnable@iastate.edu

1 **S1. Comparing the distributions of green contrast intensity between the greenhouse and** 2 **field images**

3 Kernel density estimates of green contrast intensities, i.e. $(2G - R - B)/\sqrt{6}$ distributions for the field background for field
4 background pixels, field-grown plant pixels and greenhouse plant pixels are shown in Figure S1. From the figure, we see
5 that although the green contrast density of greenhouse pixels is different from that of field-grown plant pixels, both densities
6 deviate substantially from the distribution for field background pixels. The green contrast intensities for field-grown plant pixels
7 tend to be much closer to the green contrast intensity distribution for greenhouse plant pixels than to the distribution for field
8 background pixels. Thus, a classifier built on the greenhouse plant pixels and field background pixels is able to separate the
9 field-grown plants from background.

10 **S2. Comparing segmentation methods**

11 We compare the thresholding segmentation method (using green-contrast intensity), K-means clustering of field-image pixel
12 intensities, and our proposed neural network method for each of three neighborhood sizes. One example photo and its
13 segmentation results can be found in Figure S2. Note that none of these methods requires expensive manual labelling for
14 preparing training data. From Figure S2 (b) and (c), we can see that K-means with larger K will lead to a better result compared
15 to small K ; however, many background pixels are falsely classified as plant pixels by K-means. Our 1×1 neural network
16 method is better than K-means; however, the 1×1 neural network method is visually more similar to the thresholding method

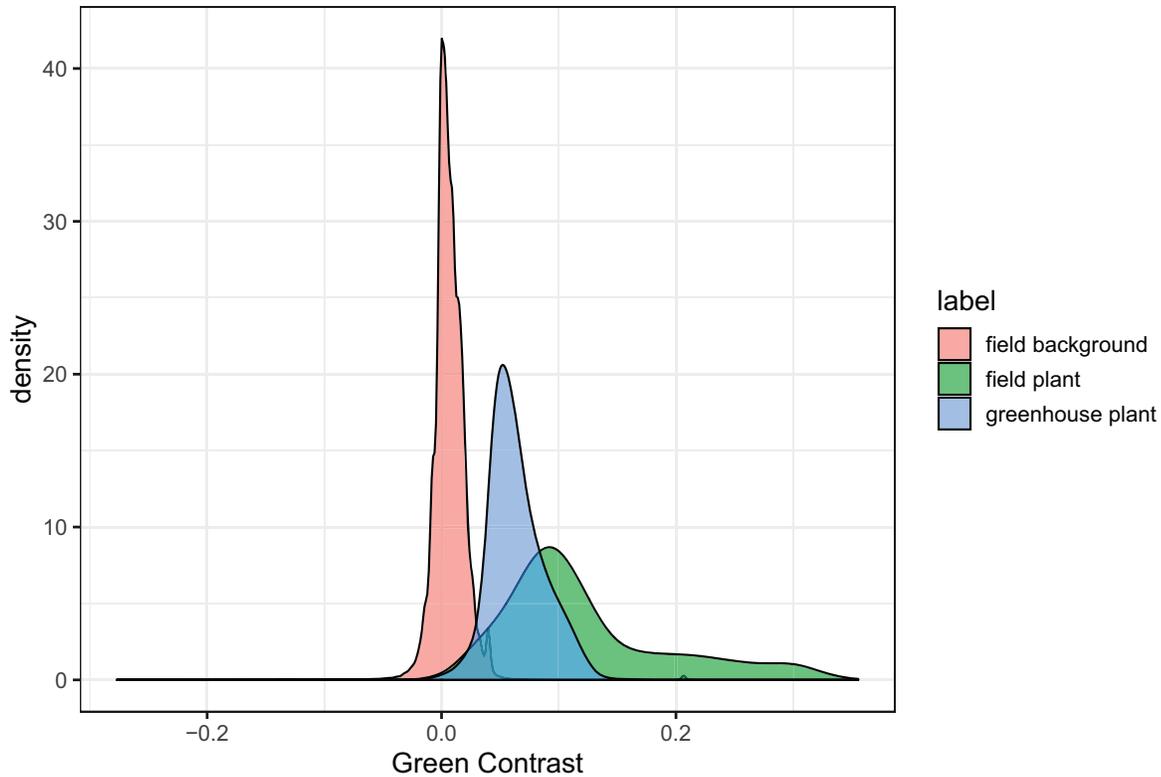


Figure S1. Distributions of green contrast intensity for greenhouse plant pixels, field plant pixels, and field background pixels.

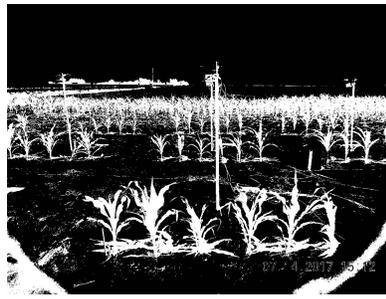
17 (see (g)) rather than the neural networks that use 3×3 and 5×5 neighborhood information (see (e) and (f)). The results from
 18 the models with 3×3 and 5×5 neighborhoods are similar. As a smaller neighborhood size reduces the computation complexity
 19 of neural network approach, we choose the 3×3 neighborhood to construct our segmentation method. Figure S3 provides
 20 the comparison of the median height estimates of 23 randomly selected images from a photo sequence using our proposed
 21 algorithm. A local smoother is used to fit each growth curve. From Figure S3, we find that the neural networks with 3×3
 22 and 5×5 neighborhoods perform similarly. The K-means algorithm fails to provide reasonable results. The neural network
 23 method using the target pixel only provides performance similarly to thresholding segmentation with threshold level 0.04. The
 24 thresholding segmentation method with threshold level 0.08 provides the smallest height.

25 **S3. Comparing different cutoff threshold values of our proposed method**

26 Figure S4 provides visual segmentation results and height measurements for three different cutoff threshold values (0.5, 0.9,
 27 and 0.95). Visually, the segmentation performs similarly across threshold values. When the cutoff threshold value is very high,
 28 say 0.95, there may be less noise, but the segmentation will provide almost the same height measurement result as for lower
 29 thresholds, which shows that our row-cut/column-cut algorithm is robust against noise.



(a) Original



(b) K-means ($K = 3$)



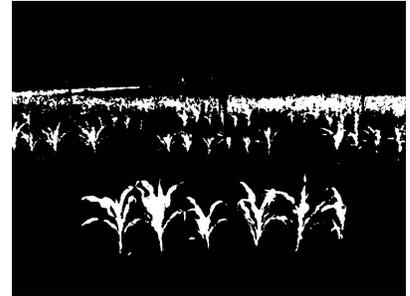
(c) K-means ($K = 9$)



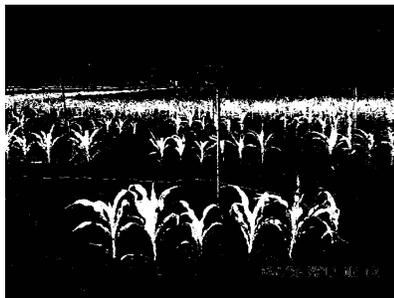
(d) Neural network (1×1)



(e) Neural network (3×3)



(f) Neural network (5×5)



(g) Green contrast thresholding (0.04)



(h) Green contrast thresholding (0.08)

Figure S2. The segmentation results of the original image in panel (a) by K-means clustering with (b): $K = 3$ and (c): $K = 9$; the proposed neural networks with (d): the target pixel only, (e): its 3×3 neighborhoods and (f): 5×5 neighborhoods; and thresholding segmentation using green-contrast intensity with threshold level (g): 0.04 and (h): 0.08.

30 **S4. Comparing segmentation results under different environmental conditions**

31 Figure S5 provides segmentation results for two sequences of photos under five different environment conditions: dawn, sunrise,
32 cloudy, foggy, and sunny, where the brightness and color scale of these images vary. From those results, we can see that our
33 algorithm successfully segments most plant pixels without much background noise, which is sufficient to estimate the height
34 accurately.

35 **S5. Evaluating the goodness of fit of the fitted growth curves**

36 To further illustrate that our proposed method leads to a stable estimation of growth curves, we evaluate goodness of fit by
37 computing the R-squared values. The boxplots of the R-squared values are presented in Figure S6, where panels (a) and (b)

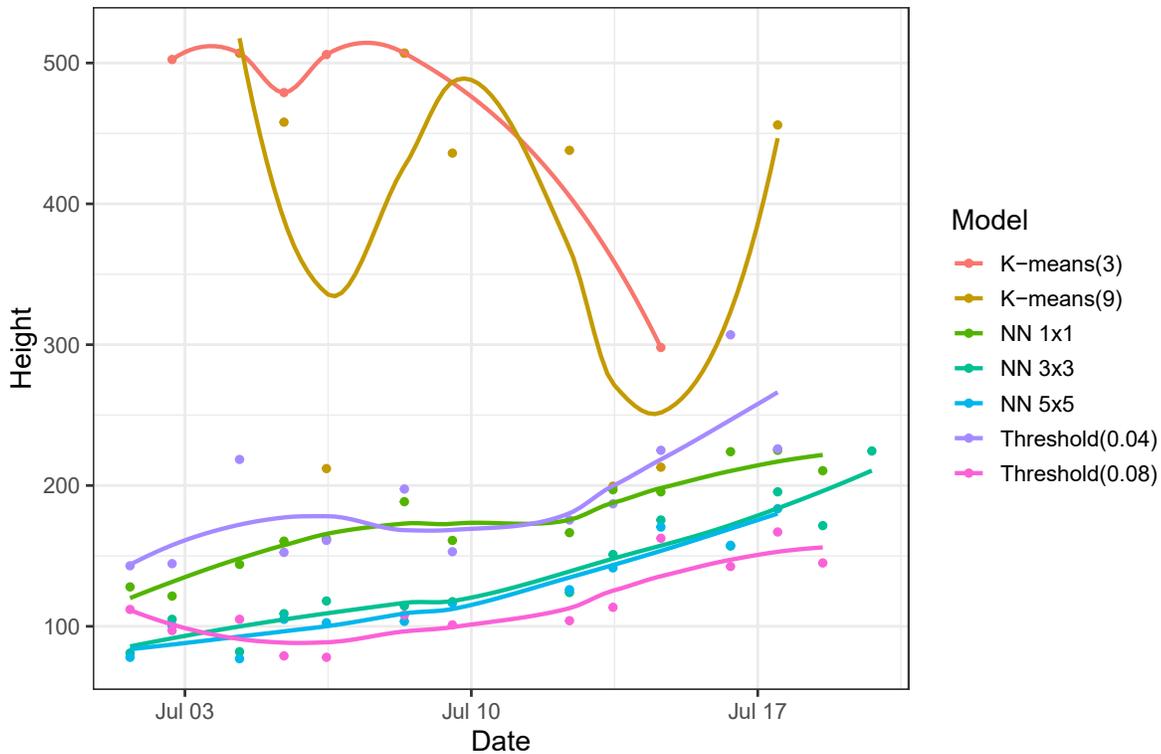


Figure S3. The comparison of the median height estimates of 23 random selected images from a photo sequence for the 8 different methods shown in Figure S2.

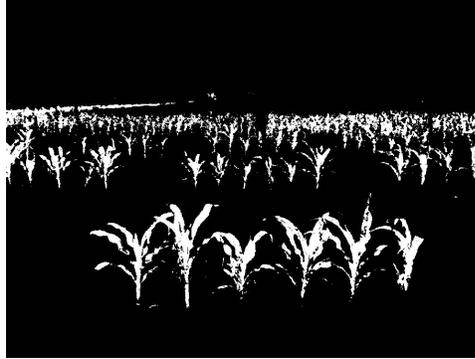
38 give the R-squared values for each camera (row) and for each plant position, respectively. We can see that our proposed
 39 non-decreasing nonparametric regression method can fit the height measurements well for most of the plants, given that most
 40 cameras have R-squared values around 0.8 or more.

41 The small R-square values for some plants from certain cameras as observed from Figure S6 in the supplementary material
 42 are due to plant death, overlapping of neighboring plants and changing weather conditions over the plant growth. Figure S7
 43 provides the extracted plant heights and the fitted growth curves for a camera (camera No.45 in Figure S6) that shows a high
 44 spread of the R-square values among the six plants. Figure S8 provides the original images on four days from this camera with
 45 the vertical lines in four different colors indicating the heights measured by our KAT4IA pipeline. The measured heights of the
 46 six plants from the four images are also highlighted in Figure S7 with the same colors as those in Figure S8.

47 Figure S8 (a) shows one image of camera 45 taken at a time with little to no wind (on 07/11/2017), and panels (b) and
 48 (c) show two images of this camera taken under windy conditions (on 07/12/2017). From those figures, we can see that due
 49 to the wind effect, the measured heights for the middle four plants under the no wind condition are much higher than those
 50 under the windy condition, especially for the heights measured from panel (b) where the plants are severely bent left by wind.
 51 Meanwhile, the height of the right-most plant is affected by the leaves of the plant left to it as seen from panel (c). Panel (d)
 52 presents a case of imaging failure which causes under-estimation of plant heights. These reasons lead to a high variation of the
 53 extracted heights and low R-square values for some cameras.



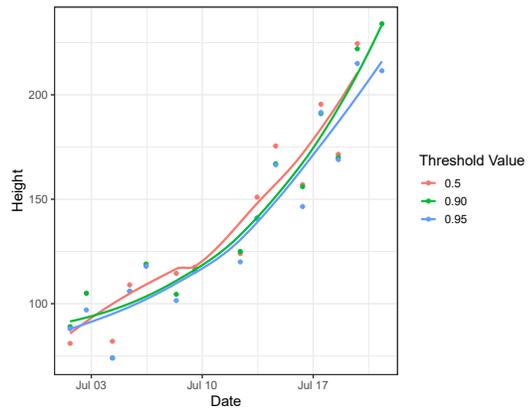
(a) threshold = 0.5



(b) threshold = 0.9



(c) threshold = 0.95



(d) compare height measurement

Figure S4. The segmented images using three cutoff threshold values are shown in (a) 0.5, (b) 0.9, (c) 0.95. The comparison of the median measured heights of 23 random selected images from a photo sequence is made among three different thresholds in (d).

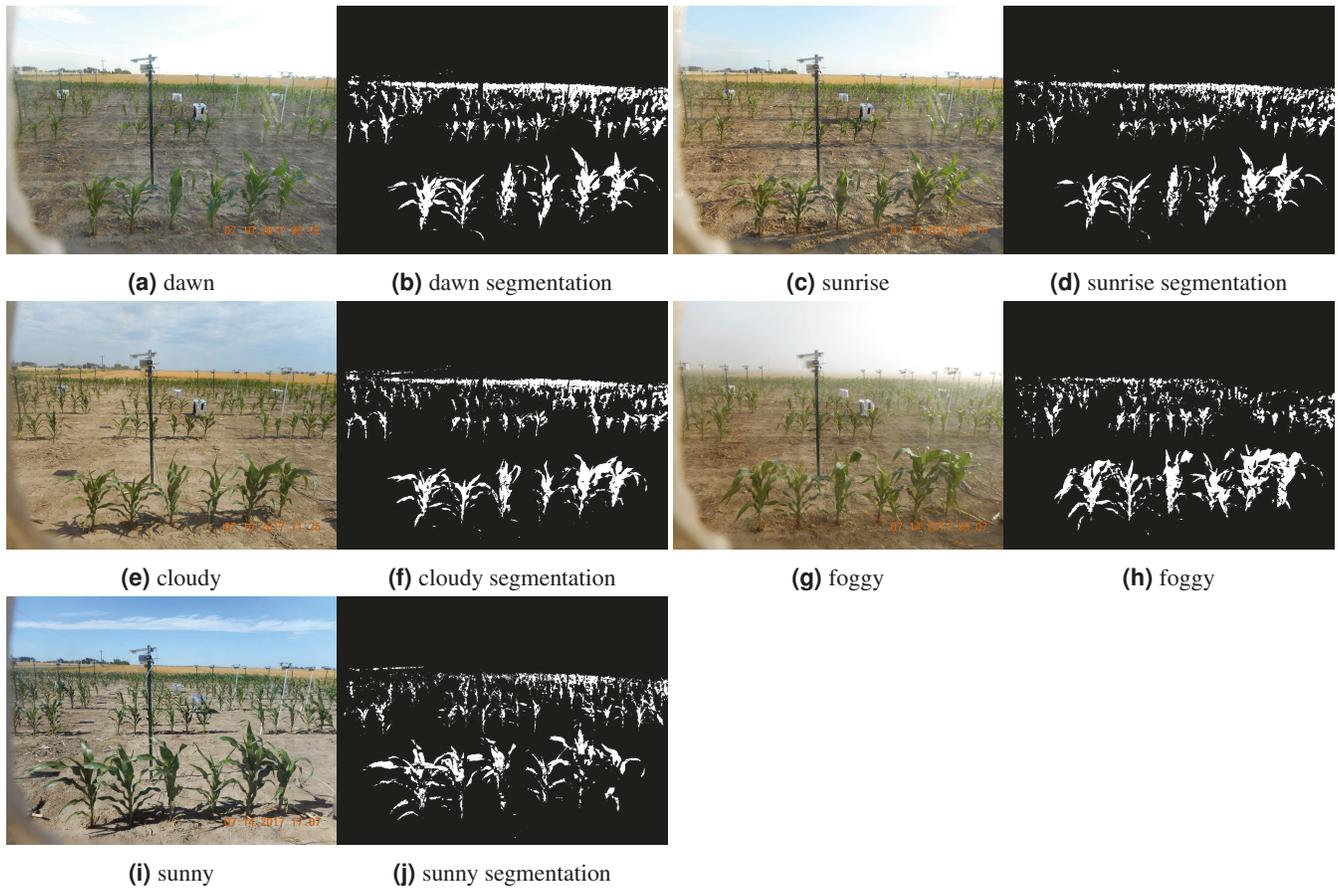


Figure S5. Segmentation results of the proposed method for images under different environmental and brightness conditions.

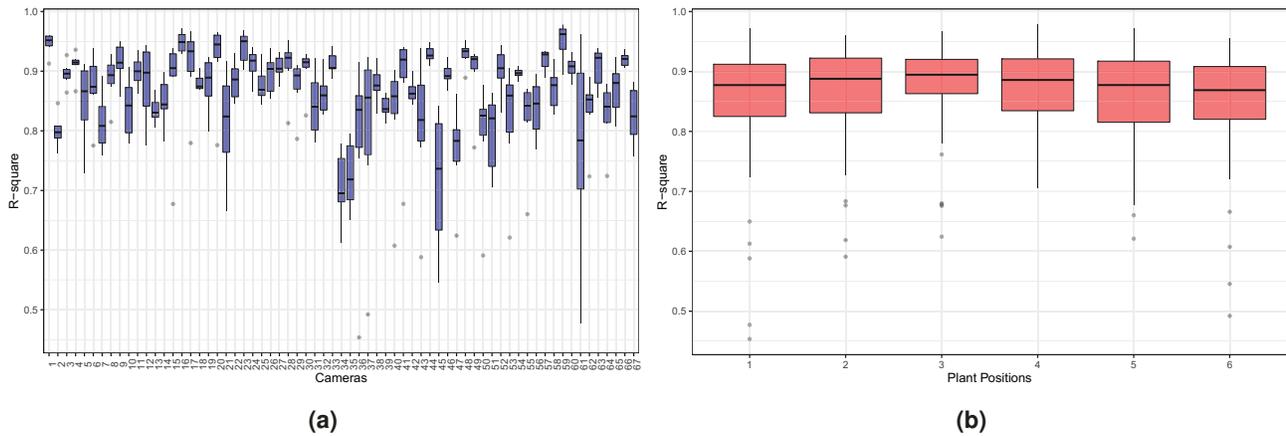


Figure S6. The R-square value for 67 cameras \times 6 plants. (a) the boxplot of R-squared values for each camera; (b) the boxplot of R-squared values for the 6 positions (i.e. left-most plants, second from the left plants, etc).

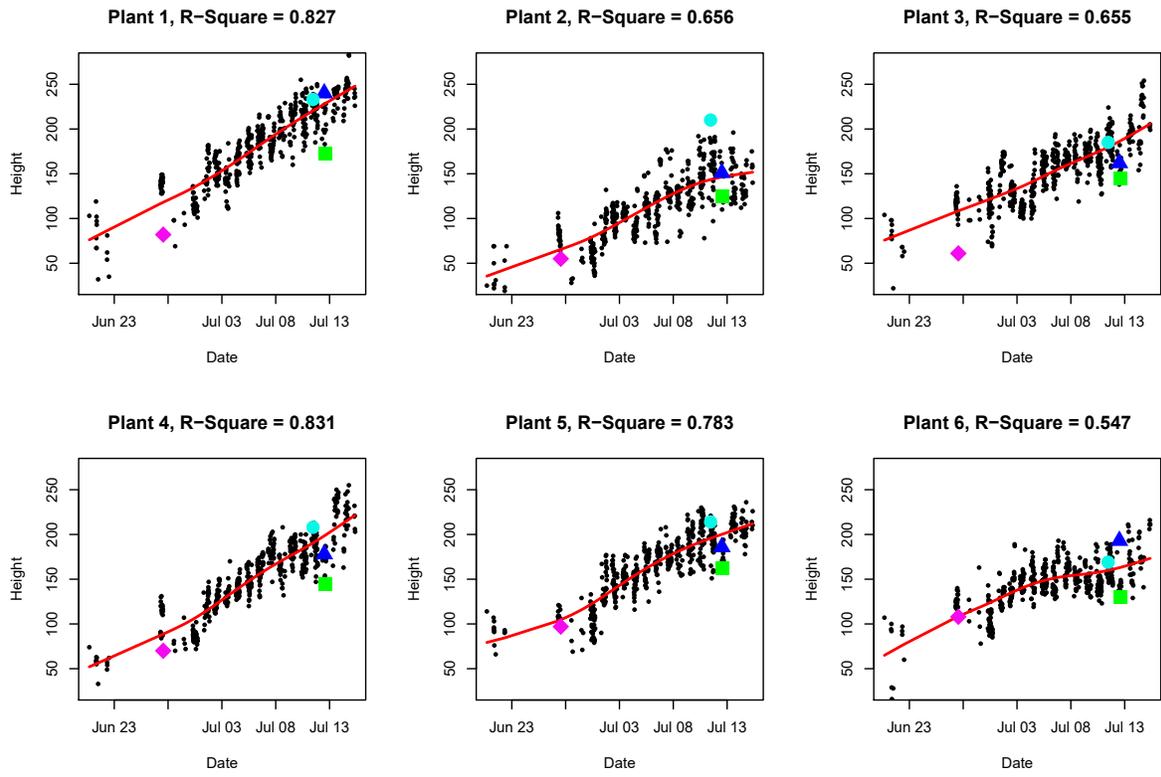


Figure S7. The extracted heights and the fitted growth curves by the proposed KAT4IA pipeline for each of the six plants in a set of images from camera No. 45. The highlighted points correspond to the cases shown in Figure R2.



(a)



(b)



(c)



(d)

Figure S8. The original images from camera No. 45 under four different conditions. The red horizontal lines and red vertical lines correspond to the results from the proposed row-cut and column-cut algorithms. The vertical lines in four colors give visualization of the extracted heights by the proposed KAT4IA pipeline.