



Research article

Phenotypic trait extraction of soybean plants using deep convolutional neural networks with transfer learning

Jason Adams¹, Yumou Qiu^{2,*}, Luis Posadas³, Kent Eskridge¹ and George Graef³

¹ Department of Statistics, University of Nebraska-Lincoln, Lincoln, NE 68583, USA

² Department of Statistics, Iowa State University, Ames, IA 50011, USA

³ Department of Agronomy and Horticulture, University of Nebraska-Lincoln, Lincoln, NE 68583, USA

* **Correspondence:** Email: yumouqiu@iastate.edu.

Abstract: High-throughput plant phenotyping systems capable of producing large numbers of images have been constructed in recent years. In order for statistical analysis of plant traits to be possible, image processing must take place. This paper considers the extraction of plant trait data from soybean images taken in the University of Nebraska-Lincoln Greenhouse Innovation Center. Using transfer learning, which utilizes the VGG16 model along with its parameters in the convolutional layers as part of our model, convolutional neural networks (CNNs) are trained to predict measurements such as height, width, and size of the plants. It is demonstrated that, by making use of transfer learning, our CNNs efficiently and accurately extract the trait measurements from the images using a relatively small amount of training data. This approach to plant trait extraction is new to the field of plant phenomics, and the superiority of our CNN-based trait extraction approach to an image segmentation-based approach is demonstrated.

Keywords: machine learning; plant phenotyping; convolutional neural networks; transfer learning

1. Introduction

The keys for quantitative analysis of plant traits are accurate and efficient collection of genetic and phenotypic data. Compared to recent advances in large-scale genetic information collection, the classical hand-measured approach for collecting plant traits is labor-intensive and inefficient. High-throughput image-based phenotyping systems have recently been built to overcome this problem. Substantial advancements have been made by engineers to enable the large-scale collection of plant images and sensor data [1–5]. A unifying objective among researchers is the accurate extraction of plant traits from images [6, 7]. While systems able to produce thousands of images per

day exist, the raw images are often only stepping stones to further analysis. For this reason, an important current area of research involves obtaining plant traits from raw images that can then be used in downstream analyses. A host of image analysis techniques from the field of computer vision are often applied to extract various plant traits, and, increasingly, machine learning algorithms are being applied to improve the accuracy of the measurements as well as the efficiency and scalability of the process [8–12].

A common approach to extracting a number of plant measurements from images first requires binary images to be produced through the process of image segmentation. Methods such as frame differencing [13], K-means [14, 15], and thresholding [16, 17] are frequently used. In [18], a neural network model was trained to classify each pixel in a maize image into either the plant class or the background class. It was demonstrated that the neural network segmentation is more accurate and robust than the traditional methods mentioned above. However, such a segmentation method based on a neural network is time consuming for high resolution images and the high-throughput phenotyping systems that produce large numbers of plant images daily for experiments.

In recent years, convolutional neural networks have become a standard approach for many image analysis tasks (see, for instance, [19–23]). They have achieved state-of-the-art results in a number of challenging areas. The main advantage of CNNs over standard feed-forward neural networks for image analysis is, by means of convolutional layers, CNNs can preserve local structural information. Intuitively, pixels are not scattered independently throughout an image. In the case of plant images, plant pixels are more likely to be next to or surrounded by other plant pixels while background pixels are more likely to be surrounded by other background pixels. Because of the preservation of local information, CNNs are thus able to learn to recognize quite complicated features with far fewer parameters than would be required of a feed-forward network [24, 25].

There have been an increasing number of applications of deep neural networks to plant phenotype extraction from images in recent years. Miao, et al. [26] employed a relatively shallow convolutional neural network (CNN) for leaf counting of maize plants. Trained using a combination of real maize images from a greenhouse and simulated maize images, this network learned to accurately count maize leaves. Lu, et al. [27] used deeper CNN structures to count the number of tassels on maize plants in an unconstrained field environment. Using 186 images for training and validation, several CNN structures were adapted from well-known models and retrained to enable accurate tassel counts. A multi-task CNN was trained in [28] to simultaneously identify wheat images containing spikes and spikelets as well as to localize and thus count said spikes and spikelets. Their CNN architecture makes use of residual blocks [29] and skip-training [30] to achieve near-perfect accuracy in spike and spikelet counting. Aich, et al. [31] utilized CNNs for estimating emergence and biomass of wheat plants from high-resolution aerial field images. The SegNet [32] architecture was used for soft segmentation as part of extracting both emergence and biomass, and a CNN utilizing inception blocks [33] and inception-residual blocks [34] was trained for each of their desired traits. In those works, CNNs with millions of parameters were constructed and estimated on limited training data. However, such large networks trained on a small amount of data may not be stable or accurate for trait prediction.

As a well-trained CNN requires a large amount of training data which may not be available in many applications, transfer learning [35] borrows the structures of pre-trained networks to solve this problem. In transfer learning, parts of a previously trained CNN are incorporated into a new network. This allows portions of a model trained for a specific task, say object classification, to be used in a

different model for a different task, say object localization. A number of the most well-known and top-performing models for many image analysis tasks have been made publicly available [19, 20, 36]. Thus not only does transfer learning reduce the need for extensive computational resources, it can also apply portions of the best CNNs to new tasks. Both the computational efficiency and improvement in task performance resulting from transfer learning have been demonstrated in a number of cases across various research domains [37–40].

This paper considers the implementation of CNNs with transfer learning to directly predict plant trait measurements from greenhouse images without segmentation. The images considered here are RGB images of soybean plants taken at the University of Nebraska-Lincoln Greenhouse Innovation Center; see Figure 1 for examples of soybean plant images. As preparing training data with accurate measurements for plant traits is both labor and time consuming, in this paper, we apply the idea of transfer learning based on the model known as VGG16 [19], which is a deep CNN originally trained for image classification on the ImageNet data set [41]. By borrowing the network structure and its parameters from the pre-trained model, we reduce the number of parameters that must be estimated in our model.

Table 1 describes the architecture of the original VGG16 model, which used 224×224 pixel RGB images as input. We pass all the soybean images to the VGG16 network and obtain the flattened vector of output from its 18th layer in Table 1. Those vectors of output are then treated as the input layer for a fully connected neural network for predicting the soybean traits. The proposed approach can be viewed as a neural network prediction method applied on the transferred images by the VGG16 model. Note that the layers 19–21 in VGG16 are not implemented in our model. More details on the implementation of the proposed method are given in the Materials and Methods section.

It will be demonstrated that our approach using less than 2000 training images can accurately produce plant trait measurements from greenhouse RGB images of soybean plants. Using available hand-measured height data, we will also demonstrate the superiority of CNN height prediction to the heights obtained from the method based on image segmentation. Those results indicate our transfer learning approach successfully implements a large neural networks trained on a relatively small sets of plant images. This advantage could be important to save time and labor in many plant phenotyping research, as different training data need to be separately prepared for different plants and experiments. With this use of transfer learning on greenhouse soybean images, our CNN method constitutes a novel approach to trait extraction from high-throughput phenotyping systems.



Figure 1. RGB Images of soybean plants in various stages of growth taken at the University of Nebraska-Lincoln Greenhouse Innovation Center.

In this study, the only hand-measurement we are able to access is the soybean plant height as measured in inches. However, the main purpose of this paper is to demonstrate the utility of machine learning to extract various trait measurements from these plant images. As such, we demonstrate the ability of CNNs to accurately predict the height, width, and what we will refer to as the size of a plant as obtained from the binary segmented images of soybean plants. These thus stand in as proxies for actual trait measurements which a CNN could be trained to obtain given hand-measured trait data. A description of how these measurements are obtained through image segmentation is given in section 2.2. We also use the hand-measured height data to train a CNN solely for height prediction to show that CNNs can be both more accurate and more efficient than the image segmentation-based method.

In summary, this paper presents two related novel contributions. First, we make use of transfer learning in training a CNN to directly extract phenotypic traits from soybean plants. That is, with no image preprocessing other than resizing, our model takes raw RGB images and predicts the plant traits on a standard measurement scale. Using this approach, we have outperformed previous state-of-the-art results in the tasks considered. Second, by incorporating part of the VGG16 model into the first layers of our model, we have been able to achieve our superior performance more efficiently relative to other, similar work in image analysis. That is, we required relatively small amount of training images and used less computation time to obtain our results. Thus this paper contributes to the field of plant phenomics by providing an efficient and accurate means of plant trait extraction via deep learning methods.

Table 1. Architecture of the original VGG16 convolutional neural network. VGG16 was trained to recognize any of 1000 different objects in the ImageNet data set, hence the output layer with 1000 units for the predicted probability of each class.

Layer Number	Layer Type	Details
1	Convolutional Layer	64 filters of size $3 \times 3 \times 3$, stride of 1, ReLU activation
2	Convolutional Layer	64 filters of size $3 \times 3 \times 64$, stride of 1, ReLU activation
3	Max Pooling Layer	2×2 kernel, stride of 2
4	Convolutional Layer	128 filters of size $3 \times 3 \times 64$, stride of 1, ReLU activation
5	Convolutional Layer	128 filters of size $3 \times 3 \times 128$, stride of 1, ReLU activation
6	Max Pooling Layer	2×2 kernel, stride of 2
7	Convolutional Layer	256 filters of size $3 \times 3 \times 128$, stride of 1, ReLU activation
8	Convolutional Layer	256 filters of size $3 \times 3 \times 256$, stride of 1, ReLU activation
9	Convolutional Layer	256 filters of size $3 \times 3 \times 256$, stride of 1, ReLU activation
10	Max Pooling Layer	2×2 kernel, stride of 2
11	Convolutional Layer	512 filters of size $3 \times 3 \times 256$, stride of 1, ReLU activation
12	Convolutional Layer	512 filters of size $3 \times 3 \times 512$, stride of 1, ReLU activation
13	Convolutional Layer	512 filters of size $3 \times 3 \times 512$, stride of 1, ReLU activation
14	Max Pooling Layer	2×2 kernel, stride of 2
15	Convolutional Layer	512 filters of size $3 \times 3 \times 512$, stride of 1, ReLU activation
16	Convolutional Layer	512 filters of size $3 \times 3 \times 512$, stride of 1, ReLU activation
17	Convolutional Layer	512 filters of size $3 \times 3 \times 512$, stride of 1, ReLU activation
18	Max Pooling Layer	2×2 kernel, stride of 2
	Flattening	Flatten array resulting from layer 18 into 1-D vector
19	Fully-connected Layer	4096 units, ReLU activation, Dropout with dropout probability 0.5
20	Fully-connected Layer	4096 units, ReLU activation, Dropout with dropout probability 0.5
21	Fully-connected Layer	Output layer, 1000 units, Softmax activation

2. Materials and methods

2.1. Data sets

There are a total of 15,223 sets of images of soybean plants in this experiment, taken from January to May 2018. Each image has a resolution of 2056×2454 pixels. There is also a subset of 2235 images for which hand-measured heights are available. The high-throughput imaging system also records millimeter-per-pixel information for each image taken that allows for converting estimated trait measurements from pixel scale to standard scales (e.g., inches in the case of height).

Of the total available image sets, two separate collections of image data were created for trait extraction. The first collection consists of 2000 images sampled at random from the total 15,223 images. As hand-measured heights were not available for all of these images, the height, width, and size measurements were obtained through segmentation (see the following subsection). This image collection will be referred to as the segmentation-obtained, or SO, collection. To predict the plant traits by CNN, this collection was split into training and testing sets. The SO training set consists of 1800 images while the SO testing set consists of the remaining 200 images from the SO collection.

The second image collection contains all 2235 images for which hand-measured heights are available. As such, this collection will be referred to as the hand-measured, or HM, collection. Similar to the SO collection, the HM collection was split into training and testing sets. The HM training set contains 1938 images while the HM testing set contains the remaining 297 images.

All images in both collections were scaled down from their original size of 2056×2454 pixels to 1024×1227 pixels by the `resize` function from the OpenCV Python library. This was done to reduce the amount of computation time needed for the methods employed.

2.2. Image segmentation and trait extraction

To obtain the height, width, and size measurements of plants via segmentation, the method of obtaining training data for image segmentation by a combination of cropping and K-means as described in [18] was employed. This process yielded a data set containing 9,325,817 total pixels as observational units. Of these, 8,857,179 were labeled as background pixels and 468,368 were plant pixels. A neural network with the same architecture as was employed for maize segmentation in [18] was then trained on these data. Once the segmentation model was trained, all images in the SO collection were segmented to binary images with 1 and 0 standing for plant and background pixels respectively. To further reduce the background noise for better trait measurements, each binary image was subjected to a morphological opening which employed the 5×5 diamond-shaped kernel matrix [42, 43]. This was accomplished using the `morphologyEx` function from the OpenCV Python library. The desired plant trait measurements could then be obtained from those segmented images. Furthermore, the HM testing set images were also segmented. This was done to allow for comparison of the segmented heights with the hand-measured heights. The Keras library in Python was used for training and prediction of the segmentation model.

As the hand-measured height was taken from the top of the pot to the top of the plant, the segmentation-obtained height in terms of pixels was calculated by taking the difference between the top of the pot and the uppermost y-position of the plant in the image. Note that the location of the pot is uniformly the same in all images under consideration. Similarly, the difference between the

right-most and the left-most x -positions of a plant yields the pixel width of the plant. The millimeter-per-pixel information of the images, which is available for both the x and y directions, were then used to scale the height and width to inches. Finally the size of the plant was obtained by simply summing the total number of plant pixels in each binary image. Again using millimeter-per-pixel information, the measurement was then scaled to square inches.

All three traits under consideration serve a practical purpose. The different experimental lines and exotic germplasm used in this study express their phenotypes along a spectrum of variation for a given trait. In this experiment, we used a set of soybean germplasm rich in genetic diversity and thus, in phenotypic variation. Of the traits under consideration, height is often used to reveal the plant growth dynamic, and its association with genetic variation. In addition to growth dynamic, plant biologists are also interested in plant architecture. In this scenario, plant width refers specifically to above-ground plant horizontal architecture. This is related to differences in petiole length, branch length, branch size, leaf length, overall plant size, among others. Similarly, plant size as measured by the millimeter-per-pixel information of the images, is a proxy for above-ground plant biomass. Hand-measurements of plant width (horizontal architecture) and size (a destructive measurement of above-ground biomass) would not only be cumbersome, but also very difficult to obtain. Therefore, the two-dimensional information provided by these measurements is significant for characterization of plant architecture.

2.3. CNN architecture and transfer learning

The Keras library in Python was used for training and prediction of the convolutional neural network. In order to predict the height, width, and size from the SO images as well as the height from the HM image collection, four separate networks were trained. These are referred to, respectively, as CNN-HSO, CNN-WSO, CNN-SSO, and CNN-HHM. While the four models learn different parameters during the training process, they all utilize the same network architecture.

As mentioned in the introduction, layers, including the pre-trained parameters, from the VGG16 model were implemented as part of our network architecture. Specifically, layers 1–18 from Table 1 served as the layers of our CNN architecture. The output from the 18th layer of the VGG16 portion of our CNN architecture was a $32 \times 38 \times 512$ array. This array was then flattened into a column vector of 622,592 units which was then fed into a fully-connected neural network with 2 hidden layers of 64 units, both of which use the ReLU activation function [25,44]. Finally, the output layer consists of only one unit and employs the linear activation function. As the plant traits are non-negative, the negative predicted values from the proposed models were replaced by 0. This fully connected portion of our neural network architecture contains 39,850,177 parameters that need to be trained for each of the four CNNs.

In addition to having the same architecture, all four CNNs were trained using the same choices of loss function, optimizer, etc. The loss function was the mean squared error, and the Adam optimizer was used with learning rate $\alpha = 0.0008$ and recommended exponential decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [44,45]. The models were each trained in batches of 128 images and ran for 100 epochs [25]. The CNN-HSO, CNN-WSO, and CNN-SSO networks were trained on the 1800 images in the SO training set and evaluated on the 200 images in the SO testing set. The CNN-HHM network was trained on the 1938 images in the HM training set and evaluated on the 297 images in the HM testing set. Note that the hyperparameters were tuned by randomly holding out 10% of the training observations as validation data and finding values that minimized the loss on the validation set of images. Once the

hyperparameters were selected, the models were retrained on the entire training data sets.

2.4. Computational time comparison

As computational time is an important consideration in image analysis tasks, the time needed to extract plant trait measurements from a 1028×1227 pixel image was measured for all four CNNs as well as for the height extraction based on segmentation. To accomplish this, one image was selected at random from the SO image collection. This image served as the input for 100 runs of each of the four CNNs. There were also 100 runs where a binary image was produced by the segmentation model and the height was calculated as described in section 2.2. Note that only the time for height extraction was calculated from the segmentation-based method as the computation time required for any of the height, width, and size measurements from segmented images is negligible.

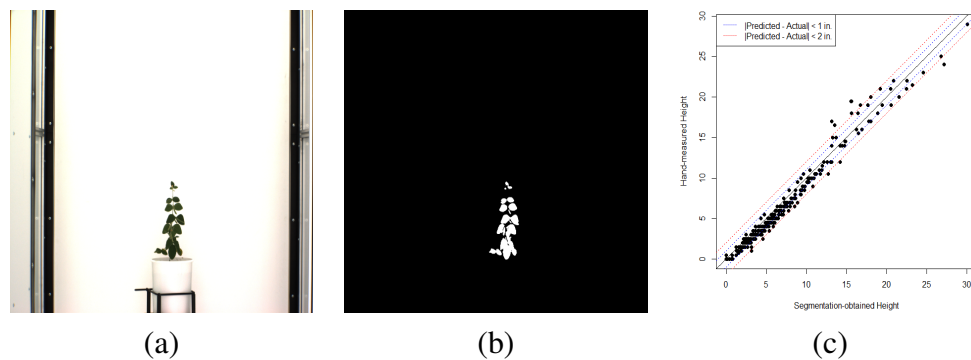


Figure 2. Panel (a) contains an RGB image of a soybean plant in the Nebraska Innovation Greenhouse while panel (b) is the resulting binary image. In panel (c), the segmentation-obtained heights are plotted against the hand-measured heights from the same images. The 45 degree line in the plots is where the heights from segmentation equal the hand-measured heights.

3. Results

3.1. Segmentation and trait extraction

Panels (a) and (b) of Figure 2 show an RGB soybean image and the corresponding binary image resulting from the proposed segmentation method. From visual inspection, our method appears to produce a well-segmented binary image. Panel (c) of Figure 2 contains the heights obtained from the binary images in the HM test set plotted against the hand-measured heights from those same images. The closer a point is to the 45 degree line seen in the plot, the smaller the discrepancy between hand-measured height and segmentation-obtained height for a given image. The mean absolute deviation between hand-measured and segmentation-obtained heights for the HM test set is 0.83. That is, for these images, the average discrepancy between measured and predicted height is only 0.83 inches. This demonstrates that our image segmentation-based approach to height extraction is viable.

3.2. CNN results

3.2.1. Results from segmentation-obtained traits

We first consider the results of the CNN-HSO, CNN-SSO, and CNN-WSO models. Figure 3(b)–(d) contain, for each of these three models, the predicted traits plotted against the traits extracted from the binary versions of the SO testing set images. Each of these plots contains a black 45 degree line representing where the predicted and observed traits are equal. In panels (b) and (c), the blue dotted lines represent deviations of 1 inch between predicted and observed heights and widths, respectively. Similarly, the red dotted lines represent deviations of 2 inches for those traits. For the size plot in panel (d), the blue dotted lines represent a deviation of 1 square inch while the red dotted lines represent a deviation of 4 square inches.

Some basic measures to assess the quality of the predictions are found in Table 2. These are the mean absolute deviation, the R^2 , and the proportions of absolute deviations falling into given regions. For the CNN-HSO and CNN-WSO, region one is less than 1 inch, region two is at least 1 inch but less than two inches, region three is at least 2 inches but less than 3 inches, and region four is at least 3 inches. As CNN-SSO predicts size, which is measured here in square inches, the regions change so that region one contains deviations less than 1 square inch, region two is at least 1 square inch but less than 4 square inches, region three is at least 4 square inches but less than 9 square inches, and region four is at least 9 square inches.

Table 2. Measures relating to the quality of trait prediction by the various CNN models. MAD represents the mean absolute deviation between predicted traits and segmentation-obtained traits while R^2 is the square of Pearson's correlation between the predicted and segmentation-obtained values. The Prop. Region rows represent the proportion of deviations falling into each region as defined in section 3.2.1.

	CNN-HSO	CNN-WSO	CNN-SSO
MAD	1.11 in.	0.8633 in.	1.51 sq in.
R^2	0.9707	0.9444	0.9723
Prop. Region 1	0.610	0.665	0.565
Prop. Region 2	0.225	0.250	0.345
Prop. Region 3	0.080	0.050	0.075
Prop. Region 4	0.085	0.035	0.015

From Table 2, the average deviations between predicted and segmentation-obtained traits are small and the R^2 values are close to 1. Further, for each model, more than 83% of all deviations fall in regions 1 and 2. For CNN-WSO and CNN-SSO, over 91% of the deviations are in regions 1 and 2. Thus, from each model, only a small proportion of the total images result in egregious deviations. Possible reasons for these deviations will be explored in the Discussion section. Overall, these measures indicate that all three models are capable of extracting the desired trait measurements accurately.

3.2.2. Results from hand-measured height

The CNN-HHM model was trained and evaluated on the HM image collection (using the HM training and testing sets, respectively). Panel (a) of Figure 3 shows that the CNN-HHM predicted

heights are quite close to the hand-measured heights on the testing data set. The lines in the plot correspond to those seen in panels (b) and (c) of Figure 3. Recall that Panel (c) in Figure 2 plots the height obtained from segmentation against the hand-measured heights for all 297 HM testing images. We compare the accuracy of CNN-HHM prediction to the method based on segmentation.

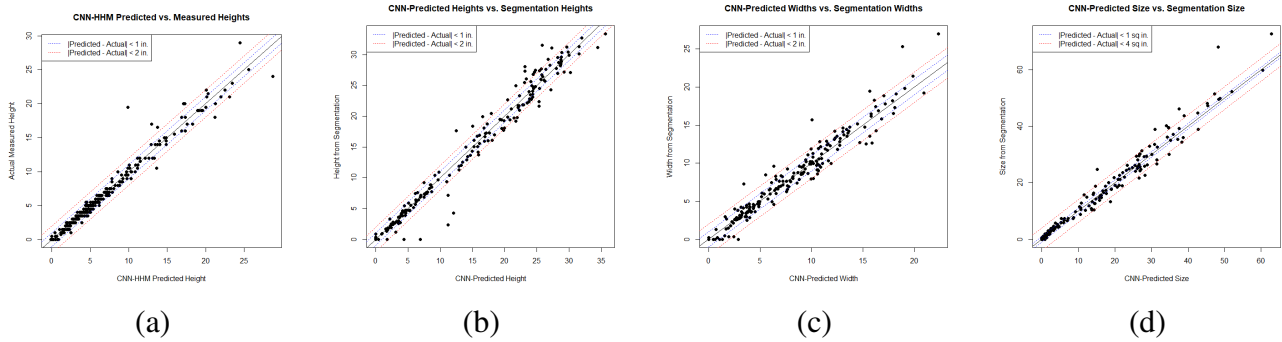


Figure 3. Plots of CNN-predicted traits from HM test images. Panel (a) contains the results of CNN-HHM, CNN-HSO results are in panel (b), panel (c) contains CNN-WSO results, and panel (d) has the predictions from CNN-SSO. The black lines in each plot are the 45 degree lines representing equality of predictions and observations. The blue and red dotted lines represent deviations of 1 and 2 inches respectively for panels (a), (b), and (c), and deviations of 1 and 4 square inches respectively for panel (d).

Table 3. Measures relating to the quality of height extraction from segmentation as well as CNN-HHM on the 297 HM testing images. See Table 2 and section 3.2.2 for explanations of the measures contained in this table.

	Segmentation-obtained	CNN-HHM
MAD	0.8348 in.	0.5236 in.
R^2	0.9758	0.9698
Prop. Region 1	0.7273	0.9057
Prop. Region 2	0.2290	0.0606
Prop. Region 3	0.0303	0.0135
Prop. Region 4	0.0134	0.0202

Table 3 contains the same measurements as those given in Table 2 for the segmentation-obtained and CNN-HHM predicted heights on the 297 HM testing set images. The regions for Table 3 are the same as those used for CNN-HSO and CNN-WSO. Here we notice that, on average, the CNN-HHM predicted heights are just over half of an inch from the hand-measured heights. This is a big improvement over the segmentation-obtained heights, whose mean absolute deviation is 0.8348 (which is still a good result). The R^2 values are comparable between the two methods with the segmentation-obtained R^2 just a little larger. The CNN-HHM also outperforms the segmentation heights in terms of the proportion of deviations within an inch of the hand-measured height. Just over 90% of CNN-HHM predictions lie within an inch of the actual height while the value is close to 73% for the segmentation-obtained heights. The one area in which the segmentation-obtained heights appear to do better than

the CNN-HHM predicted heights can be observed from the plot in Figure 3(a). It appears that even though most of the deviations in this plot are smaller than the deviations in panel (c) of Figure 2 (i.e., they are more tightly clustered around the 45 degree line), there are some exceptionally large deviations produced by the CNN-HHM predictions. As with the other three CNN models, only a small percentage of images produce egregious deviations from the hand-measured heights. These will also be investigated further in the Discussion section. Overall, it appears that CNN-HHM extracts the soybean height more accurately than the segmentation method does.

3.3. Computational time results

In addition to accuracy in trait extraction, computational time is an important consideration when comparing image analysis methods. As mentioned in the Materials and Methods, all five trait extraction methods considered above were run 100 times on the same image, and completion time, as measured in seconds, was recorded for each run. Table 4 contains the average times and standard deviations for each method as well as 95% confidence intervals for the difference in average run time between each of the CNN models and the segmentation method.

Table 4. Table of computational time results. The 95% t-intervals are for the average difference between each of the CNN models and the segmentation-based method.

Method	Mean Run Time	Standard Deviation	95% t-Intervals
Segmentation	53.35	1.7641	
CNN-HSO	4.67	0.4291	(48.32, 49.04)
CNN-WSO	4.71	0.4128	(48.29, 49.00)
CNN-SSO	4.86	0.3969	(48.14, 48.85)
CNN-HHM	4.85	0.4098	(48.15, 48.86)

It is clear from Table 4 that all of the CNN models extract plant traits much faster than the segmentation method. This demonstrates another significant advantage that the CNN models have over the segmentation approach to plant trait extraction. While trait extraction is not instantaneous for any of the CNN models, an extraction time of approximately 5 s per image should be sufficiently fast for most purposes.

4. Discussion

As mentioned in the Results section, the CNN models sometimes make predictions that are far from the observed trait measurements. In particular, some of the height predictions from CNN-HHM deviated especially far from the hand-measured height values. To investigate why this might be the case, the six images producing discrepancies between predicted and actual height of at least 3 inches are shown in Figure 4. The black line on the left of each image is the height as measured in the greenhouse, and the red line on the right of the image is the height predicted by CNN-HHM. Listed below each image is the absolute height discrepancy in inches between measured and predicted values.

Interestingly, the image in panel (a) of Figure 4 also produces the largest discrepancy between measured height and segmentation-obtained height. While the segmentation-obtained height only produces a discrepancy of 3.92 inches compared to the discrepancy of 9.6 inches produced by the

CNN-HHM height, this image presents an instructive case. Observe that the plant in panel (a) of Figure 4 is not standing erect but curves instead. This pattern is expected when growing soybean plants under greenhouse conditions, but it is also undesirable. The vine-like growing pattern is considered to be an artifact where temperature and light, both natural and artificial, play a role in internode elongation of the main stem of the plant [46–48]. This is the reason that bamboo sticks are seen in many of the images as the plants were attached to the sticks in an attempt to keep them as upright as possible. So in panel (a) of Figure 4 the plant was made to stand erect when the height measurement was taken. It can also be observed from this image that CNN-HHM appears to be confused by the curve in the stem as it ends its height prediction at the point where the plant posture is no longer erect. This could be due to not having enough examples of images such as this one in the HM training set. This seems likely as most of the plants with fast elongating internodes were attached to bamboo sticks before they elongated in excess, and thus there are not many examples of vine-like growing plants this size that have not yet been straightened out.

The bamboo sticks themselves appear to be the primary cause of discrepancy in panels (b), (c) and (e) of Figure 4. CNN-HHM is evidently confused about where the plant ends towards the top of the bamboo sticks. As for panels (d) and (f) of Figure 4, if the mistakes are due to CNN-HHM, it is difficult to see what those might be. It could actually be the case that the measurements were taken improperly. In panel (d) of Figure 4, the measured height seems to be too tall. We believe the most likely cause for the discrepancy in this image is simply a data-entry error. At least to the naked eye, CNN-HHM appears to give the more accurate measurements in panels (d) and (f) of Figure 4.

Based on these large discrepancies, one potential means of reducing large errors made by the CNN would be to ensure a consistent approach to measurement in difficult cases. Surely some cases, such as panel (a) of Figure 4, will be too infrequent for sufficient examples to exist in the set of training images. In other cases, inconsistency in measurement procedure can make training more difficult. If similar plants are measured differently, the CNN may find some trivial way to distinguish between them that it should not use in predicting the measurements.

In relation to the computation time results, it is important to note that the segmentation procedure employed depends on a neural network, and this is the primary reason that extracting the height from a single image takes so long. For more traditional approaches to segmentation such as those mentioned in the introduction, the height extraction time would surely be much faster. It is likely that height extraction time using frame differencing, K-means, or a thresholding procedure would be even shorter than the approximately 5 s average time obtained by the four CNN models. However, the primary reason this was not looked into further was our inability to obtain acceptable segmentation results on the soybean images using any of these methods. The presence of bamboo sticks in many of the images make finding a usable reference image for frame differencing difficult. We were also unable to find configurations of K-means or thresholding methods that could distinguish the dark background from the dark soybean plants.

The results presented here lead to the conclusion that convolutional neural networks and transfer learning can be trained to efficiently and accurately extract plant height, width and size from images. We have considered more sophisticated traits which also demonstrate the utility of the proposed method. Miao, et al. [26] employs a CNN for obtaining leaf counts from maize images taken in the same greenhouse as our soybean plant images. By using the same network architecture and hyperparameter choices described for the CNNs in the materials and methods section (the sole

exception is that the model was trained over 50 epochs rather than 100), we have been able to improve on the maize leaf counting results presented in [26]. In addition to the deeper network architecture, we also used a larger number of images for training. Comparing results between test sets, they reported an R^2 value of 0.74 and a root mean squared error (RMSE) of 1.33 while we obtained an R^2 of 0.88 and RMSE of 1.10. This encouraging result supports the conclusion that CNNs and transfer learning can be successfully applied to plant trait extraction for a variety of both plants and traits.

Additionally, though the CNN models given here were all trained on greenhouse images containing only a single plant per image, some modifications could be made to allow for trait extraction from images containing multiple plants. Building on the data collection and segmentation from [18, 49] presents a method to isolate, segment, and compute (segmentation-based) heights for maize plants in field images. In addition, techniques such as R-CNNs and the YOLO algorithm [50–52] have been successful in identifying and localizing objects in images. Combined with a reliable method to isolate plants in more varied images, a transfer learning CNN trait extraction algorithm would likely be much more widely applicable based on labelled features from several hundreds of those separated field-plant images as training data. This paper demonstrates the potential of CNNs, combined with transfer learning, to successfully and efficiently extract phenotypic traits from plant images. It serves as a useful starting point to building more sophisticated models that will allow for trait extraction from a greater number of plant species in more varied settings.

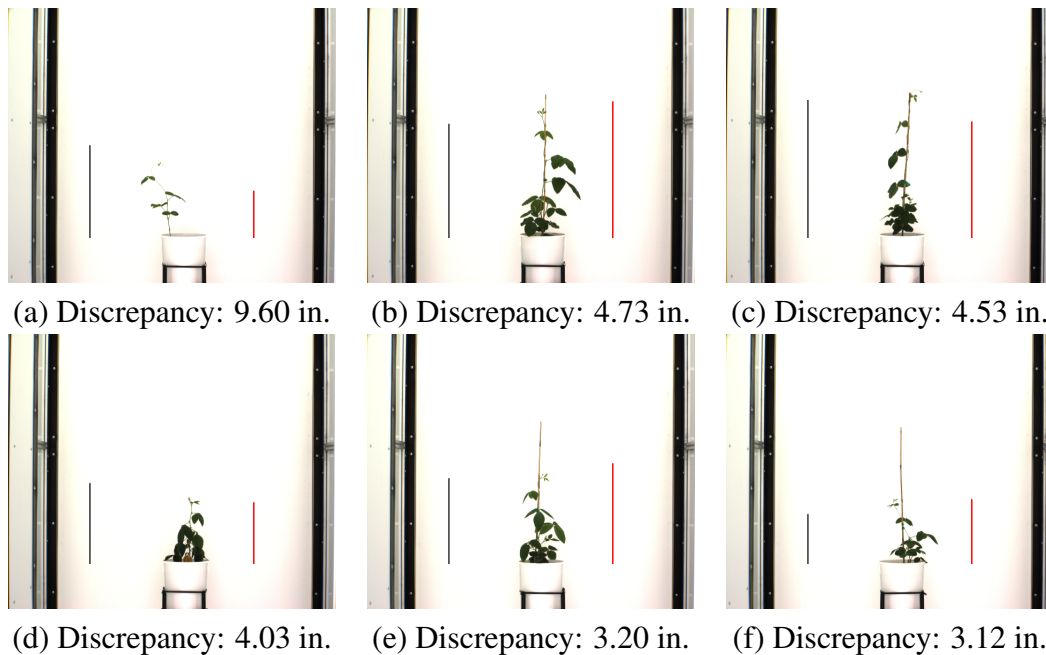


Figure 4. Images with discrepancies (the number below each image) between hand-measured and CNN-HHM predicted heights over 3 inches. The black line on the left side of each image represents the hand-measured height while the red line represents the CNN-HHM predicted height.

Acknowledgements

This project is based on research that was partially supported by the Nebraska Agricultural Experiment Station with funding from the Hatch Act through the USDA National Institute of Food and Agriculture. Other financial support was provided by grants from the Nebraska Soybean Board. We would like to thank the Plant Phenotyping Committee at UNL, and in particular IANR Deans Tala Awada and Hector Santiago, for the guidance and expertise provided.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Chéné Y, Rousseau D, Lucidarme P, et al. (2012) On the use of depth camera for 3D phenotyping of entire plants. *Comput Elect Agr* 82: 122–127.
2. McCormick RF, Truong SK, Mullet JE, (2016) 3D sorghum reconstructions from depth images identify QTL regulating shoot architecture. *Plant physiol* 172: 823–834.
3. Xiong X, Yu L, Yang W, et al. (2017) A high-throughput stereo-imaging system for quantifying rape leaf traits during the seedling stage. *Plant Methods* 13: 1–17.
4. Peñuelas J and Filella I, (1998) Visible and near-infrared reflectance techniques for diagnosing plant physiological status. *Trends Plant Sci* 3: 151–156.
5. Lin Y, (2015) LiDAR: An important tool for next-generation phenotyping technology of high potential for plant phenomics? *Comput Elect Agr* 119: 61–73.
6. Fahlgren N, Gehan MA, Baxter I, (2015) Lights, camera, action: high-throughput plant phenotyping is ready for a close-up. *Curr Opin Plant Biol* 24: 93–99.
7. Miller ND, Parks BM, Spalding EP, (2007) Computer-vision analysis of seedling responses to light and gravity. *Plant J* 52: 374–381.
8. Miao C, Yang J, Schnable JC, (2019) Optimising the identification of causal variants across varying genetic architectures in crops. *Plant Biotech J* 17: 893–905.
9. Xavier A, Hall B, Casteel S, et al. (2017) Using unsupervised learning techniques to assess interactions among complex traits in soybeans. *Euphytica* 213: 1–18.
10. Habier D, Fernando RL, Kizilkaya K, et al. (2011) Extension of the Bayesian alphabet for genomic selection. *BMC Bioinform* 12: 1–12.
11. Gage JL, Richards E, Lepak N, et al. (2019) In-field whole-plant maize architecture characterized by subcanopy rovers and latent space phenotyping. *Plant Phenome J* 2: 1–11.
12. Wu H, Wiesner-Hanks T, Stewart EL, et al. (2019) Autonomous detection of plant disease symptoms directly from aerial imagery. *Plant Phenome J* 2: 1–9.
13. Choudhury SD, Bashyam S, Qiu Y, et al. (2018) Holistic and component plant phenotyping using temporal image sequence. *Plant Methods* 14: 1–21.
14. Johnson RA and Wichern DW, (2002) *Applied Multivariate Statistical Analysis*. Prentice Hall Upper Saddle River, NJ.

15. Klukas C, Chen D, Pape JM, (2014) Integrated analysis platform: an open-source information system for high-throughput plant phenotyping. *Plant Physiol* 165: 506–518.
16. Hartmann A, Czauderna T, Hoffmann R, et al. (2011) HTPPheno: an image analysis pipeline for high-throughput plant phenotyping. *BMC Bioinform* 12: 1–9.
17. Ge Y, Bai G, Stoerger V, et al. (2016) Temporal dynamics of maize plant growth, water use, and leaf water content using automated high throughput RGB and hyperspectral imaging. *Comput Elect Agr* 127: 625–632.
18. Adams J, Qiu Y, Xu Y, et al. (2020) Plant segmentation by supervised machine learning methods. *Plant Phenome J* 3: e20001.
19. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.
20. Krizhevsky A, Sutskever I, Hinton GE, (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Infor Process Syst* 25: 1097–1105.
21. Zhu X and Ramanan D, (2012) Face detection, pose estimation, and landmark localization in the wild. *2012 IEEE Confer Comput Vision Pattern Recognit*: 2879–2886.
22. Gatys LA, Ecker AS, Bethge M, (2016) Image style transfer using convolutional neural networks. *Process IEEE Confer Comput Vision Pattern Recognit*: 2414–2423.
23. Liang Z, Powell A, Ersoy I, et al. (2016) CNN-based image analysis for malaria diagnosis. *2016 IEEE Int. Confer Bioinform Biomed (BIBM)*: 493–496.
24. LeCun Y, Bengio Y, Hinton G, (2015) Deep learning. *Nature* 521: 436–444.
25. Goodfellow I, Bengio Y, Courville A, et al. (2016) *Deep Learning*. MIT Press, Cambridge.
26. Miao C, Hoban TP, Pages A, et al. (2019) Simulated plant images improve maize leaf counting accuracy. *BioRxiv*: 706994.
27. Lu H, Cao Z, Xiao Y, et al. (2017) TasselNet: counting maize tassels in the wild via local counts regression network. *Plant Methods* 13: 1–17.
28. Pound MP, Atkinson JA, Wells DM, et al. (2017) Deep learning for multi-task plant phenotyping. *Process IEEE Int Confer Comput Vision Workshops*: 2055–2063.
29. He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. *Process IEEE Confer Comput Vision Pattern Recognit*: 770–778.
30. Orhan AE and Pitkow X, (2017) Skip connections eliminate singularities. *arXiv:1701.09175*.
31. Aich S, Josuttis A, Ovsyannikov I, et al. (2018) Deepwheat: Estimating phenotypic traits from crop images with deep learning. *2018 IEEE Winter Confer Appl Comput Vision (WACV)*: 323–332.
32. Badrinarayanan V, Kendall A, Cipolla R, (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Machine Intell* 39: 2481–2495.
33. Szegedy C, Liu W, Jia Y, et al. (2015) Going deeper with convolutions. *Process IEEE Confer Comput Vision Pattern Recognit*: 1–9.
34. Szegedy C, Ioffe S, Vanhoucke V, et al. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. *Process AAAI Confer Artif Intell* 31.
35. Pan SJ and Yang Q, (2009) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22: 1345–1359.
36. LeCun Y, Bottou L, Bengio Y, et al. (1998) Gradient-based learning applied to document recognition. *Process IEEE* 86: 2278–2324.

37. Shin H-C, Roth HR, Gao M, et al. (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans Med Imag* 35: 1285–1298.
38. Han D, Liu Q, Fan W, (2018) A new image classification method using CNN transfer learning and web data augmentation. *Expert Syst Appl* 95: 43–56.
39. Akcay S, Kundegorski ME, Willcocks CG, et al. (2018) Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE Trans Infor Forensics Security* 13: 2203–2215.
40. Xie M, Jean N, Burke M, et al. (2016) Transfer learning from deep features for remote sensing and poverty mapping. *Procee AAAI Confer Artif Intell* 30.
41. Deng J, Dong W, Socher R, et al. (2009) Imagenet: A large-scale hierarchical image database. *2009 IEEE Confer Comput Vision Pattern Recognit*: 248–255.
42. Shapiro L, (1992) *Computer Vision and Image Processing*: Academic Press.
43. Davies ER, (2012) *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Academic Press.
44. Nielsen MA, (2015) *Neural Networks and Deep Learning*. Determination Press, San Francisco, CA.
45. Kingma DP and Ba J, (2014) Adam: A method for stochastic optimization. *arXiv:14126980*.
46. Zhang L, Allen Jr LH, Vaughan MM, et al. (2014) Solar ultraviolet radiation exclusion increases soybean internode lengths and plant height. *Agric For Meteorol* 184: 170–178.
47. Allen Jr LH, Zhang L, Boote KJ, et al. (2018) Elevated temperature intensity, timing, and duration of exposure affect soybean internode elongation, mainstem node number, and pod number per plant. *Crop J* 6: 148–161.
48. Downs J and Thomas JF, (1990) Morphology and reproductive development of soybean under artificial conditions. *Biotronics* 19: 19–32.
49. Guo X, Qiu Y, Nettleton D, et al. (2020) Automatic traits extraction and fitting for field high-throughput phenotyping systems. *bioRxiv*.
50. Girshick R, (2015) Fast r-CNN. *Procee IEEE Int Confer Comput Vision*: 1440–1448.
51. Ren S, He K, Girshick R, et al. (2016) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Machine Intell* 39: 1137–1149.
52. Redmon J, Divvala S, Girshick R, et al. (2016) You only look once: Unified, real-time object detection. *Procee IEEE Confer Comput Vision Pattern Recognit*: 779–788.

Additional information

Python code for the implementation of the proposed method, trained models, and example images are available at <https://github.com/jasonradams47/SoybeanTraitPrediction>.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)