1 Root Anatomy based on Root Cross-Section Image 2 Analysis with Deep Learning

3 Chaoxin Wang^a, Xukun Li^a, Doina Caragea^{a,*}, Raju Bheemanahallia^b, S.V.
4 Krishna Jagadish^b

5 ^a*Department of Computer Science, Kansas State University, Manhattan, KS, USA*

6 ^b*Department of Agronomy, Kansas State University, Manhattan, KS, USA*

7 **Abstract** Aboveground plant efficiency has improved significantly in
8 recent years, and the improvement has led to a steady increase in global food
9 production. The improvement of belowground plant efficiency has potential to
10 further increase food production. However, belowground plant roots are harder
11 to study, due to inherent challenges presented by root phenotyping. Several
12 tools for identifying root anatomical features in root cross-section images
13 have been proposed. However, the existing tools are not fully automated and
14 require significant human effort to produce accurate results. To address this
15 limitation, we use a fully automated approach, specifically, the Faster Region-
16 based Convolutional Neural Network (Faster R-CNN), to identify anatomical
17 traits in root cross-section images. By training Faster R-CNN models on

*Corresponding author

Email addresses: cwang16@ksu.edu (Chaoxin Wang), xukun@ksu.edu (Xukun Li),
dcaragea@ksu.edu (Doina Caragea), rajubr@ksu.edu (Raju Bheemanahallia),
kjagadish@ksu.edu (S.V. Krishna Jagadish)

18 root cross-section images, we can detect objects such as root, stele and late
 19 metaxylem, and predict rectangular bounding boxes around such objects.
 20 Subsequently, the bounding boxes can be used to estimate the root diameter,
 21 stele diameter, late metaxylem number, and average diameter. Experimental
 22 evaluation using standard object detection metrics, such as intersection-over-
 23 union and mean average precision, has shown that the Faster R-CNN models
 24 trained on rice root cross-section images can accurately detect root, stele
 25 and late metaxylem objects. Furthermore, the results have shown that the
 26 measurements estimated based on predicted bounding boxes have small root
 27 mean square error when compared with the corresponding ground truth values,
 28 suggesting that Faster R-CNN can be used to accurately detect anatomical
 29 features. A webserver for performing root anatomy using the Faster R-CNN
 30 models trained on rice images is available at <https://rootanatomy.org>, together
 31 with a link to a GitHub repository that contains a copy of the Faster R-CNN
 32 code. The labeled images used for training and evaluating the Faster R-CNN
 33 models are also available from the GitHub repository.

34 **Keywords:** Image Analysis, Deep Learning, Object Detection, Faster
 35 R-CNN, Root Anatomy

36 1. Introduction

37 The crop scientific community has made significant strides in increasing
 38 global food production through advances in genetics and management, with
 39 majority of the progress achieved by improving aboveground plant efficiency
 40 [1, 2, 3]. The belowground plant roots, which provide water and nutrients
 41 for plant growth, are relatively less investigated. This is primarily because of
 42 the difficulty in accessing the roots, and the complexity of phenotyping root
 43 biology and function [4, 5]. Hence, root potential has largely been untapped
 44 in crop improvement programs [4, 5]. Over the past decade, different root
 45 phenotyping approaches have been developed for studying root architecture,
 46 including basket method for root angle [6], rhizotron method for tracking root
 47 branching, architecture and growth dynamics [7], shovelomics, a.k.a., root
 48 crown phenotyping [8], among others. Recent advances in magnetic resonance
 49 imaging and X-ray computed tomography detection systems have provided
 50 the opportunity to investigate root growth dynamics in intact plants at high
 51 temporal frequency [9, 10, 11, 12, 13]. However, each of these techniques
 52 comes with a range of inherent biases or limitations (such as artificial plant
 53 growth conditions), with none of the techniques currently available clearly
 54 standing out as a promising “blanket fit” approach [14, 15, 16]. Recent non-

55 destructive technologies, such as X-ray computed tomography, are extremely
56 expensive, and thus beyond the reach of common crop improvement programs,
57 in addition to not having the bandwidth to capture large genetic diversity.

58 Machine learning is an area of artificial intelligence, focused on models
59 that can automatically infer patterns from existing data, without human
60 intervention [17]. In supervised machine learning, data in the form of training
61 instances are provided as input, and the models learn patterns that can be
62 used to make predictions on new unseen data. Machine learning approaches
63 have been used successfully to address a wide variety of bioinformatics and
64 computational biology problems relevant to crop sciences, including prediction
65 of gene functions in plants [18], discovery of single nucleotide polymorphisms
66 (SNP) in polyploid plants [19], subcellular localization [20], genomic selection
67 for plant breeding [21], high-throughput plant phenotyping based on image
68 analysis [22], prediction of biomass [23]. Furthermore, applications of advanced
69 deep neural networks to challenging problems in crop analysis have led to
70 state-of-the-art results that outperform the results of traditional machine
71 learning and image analysis techniques approaches as reviewed in [24].

72 Most relevant to this work, machine learning, in general, and deep neural
73 networks (a.k.a., deep learning), in particular, are expanding the ability to

74 accurately predict a plant phenotype [25, 26, 27, 28, 29, 30, 31, 32, 33]. These
 75 technological advances have enabled researchers to capture a wide range of
 76 genetic diversity, a task which has been hardly possible in the past, given
 77 the amount of time and effort involved in manual analysis. Several recent
 78 studies have used deep learning approaches for identifying and quantifying
 79 aboveground plant traits, such as the number of leaves in rosette plants, based
 80 on high-resolution RBF images [29, 30, 31]. Other investigations have focused
 81 on identifying plant diseases [34] or on stress phenotyping [26].

82 Furthermore, several prior studies have focused on data-driven approaches
 83 and tools for belowground plant phenotyping, including identifying and quan-
 84 tifying root morphological parameters, such as changes in root architecture,
 85 or branching and growth [35, 36, 37, 38]. Such approaches rely on standard
 86 image analysis techniques as opposed to state-of-the-art deep learning.

87 Both root morphological and anatomical traits are important in relation
 88 to the efficiency of soil moisture absorption by the root system. Large genetic
 89 variation in root related traits has positioned rice to uptake water and increase
 90 yields under a range of ecological conditions, including flooded and dryland
 91 conditions [39]. Root anatomical traits such as nodal root diameter (RD) [40],
 92 late metaxylem diameter (LMXD) and number (LMXN) [41, 42, 43], and stele

diameter (SD) and its proportion to root diameter (SD:RD) [44] have been proposed as key traits for optimized acquisition of water and productivity under water-limited conditions [40]. Thin SD:RD has been used as a surrogate measure of cortex tissue area/width, which helps in the improvement of water flow and retention in vascular tissue [45, 44]. Late metaxylem number and diameter along the root influence the hydraulic conductivity [41, 44]. These parameters mentioned above help to determine effective water use throughout the crop growth period [46, 43].

Innovations in image acquisition technologies have made it possible to gather relatively large sets of root cross-section images, enabling studies on root anatomy. Several approaches and tools for quantifying root anatomical variation based on cross-section images have been proposed in recent years [47, 48, 49]. However, the existing tools are only partially automated, as they require user input and fine-tuning of the parameters for each specific image or for a batch of images. Fully automated tools exist for the analysis of hypocotyl cross-sections (i.e., the region in between seed leaves and roots) in the context of secondary growth [50, 51], but they are not directly applicable to the analysis of root cross-section images. Thus, there is a pressing need for automated root cross-section image analysis tools that can be used to

112 perform root anatomy at a low cost.

113 To address this limitation, we have taken advantage of recent advances in
 114 deep learning and image analysis, and used a state-of-the-art, fully-automated
 115 deep learning approach, the Faster R-CNN network [52], to identify and
 116 quantify root anatomical parameters indicative of physiological and genetic
 117 responses of root anatomical plasticity in field crops. Specifically, as a proof-
 118 of-concept, we have focused on the following parameters: root diameter (RD),
 119 stele diameter (SD), late metaxylem diameter (LMXD) and late metaxylem
 120 number (LMXN), which were found important in relation to water-deficit
 121 stress in our prior work [44, 53]. A graphical illustration of these parameters
 122 is shown in Figure 1.

123 The existing Faster R-CNN model was trained on rice root cross-section
 124 images. The trained model was used to detect objects of interest in a root
 125 cross-section image (i.e., root, stele and late metaxylem), together with their
 126 corresponding bounding boxes. Subsequently, the bounding boxes were used
 127 to estimate anatomical parameters such as RD, SD, LMXD, LMXN. The
 128 Faster R-CNN model generalizes well to unseen images, thus eliminating the
 129 need for the end-user to hand-draw a stele border or manually choose or
 130 correct the metaxylem cells, tasks that are time-consuming, and also prone

Figure 1: Root anatomical traits. (Top) Root cross-section with highlighted *root diameter* and *stele*. Image taken at 50x magnification. (Bottom) Enlarged stele with highlighted *stele diameter*, and *late metaxylem diameter*. The *late metaxylem number* is also a trait of interest. The image was taken at 100x magnification.

131 to noise and errors.

132 To summarize, our main contributions are as follows:

- 133 • We have used the Faster R-CNN network trained on root cross-section
- 134 images to detect root, stele and late metaxylem objects, and their
- 135 corresponding bounding boxes.
- 136 • We have investigated the Faster R-CNN model with respect to the
- 137 number of instances needed to accurately detect objects of interest, and
- 138 their corresponding bounding boxes.

- 139 • We have evaluated the ability of the predicted bounding boxes to
140 produce accurate estimates for anatomical properties, and performed
141 error analysis to identify sources of errors.
- 142 • We have identified advantages and disadvantages of Faster R-CNN
143 approach for root anatomy by comparison with existing approaches for
144 this task.

145 2. Materials and Methods

146 While there are many anatomical traits that can be identified, and mea-
147 sured or counted (e.g., RootScan outputs more than 20 anatomical parame-
148 ters), as a proof-of-concept, we have focused on measuring the root diameter
149 (RD), stele diameter (SD), and late metaxylem diameter (LMXD), and count-
150 ing the number of late metaxylem inside the stele (LMXN). Our choice was
151 motivated by studies by Kadam et al. [44, 53], who showed the importance
152 of these traits in relation to water-deficit stress, and provided the ground
153 truth dataset for our study. The tasks that we target can be achieved with
154 modern object detection techniques, such as Faster R-CNN, as described
155 below. In addition to the traits of interest (RD, SD, LMXD and LMXN),
156 other traits can be estimated based on the objects detected with our trained

157 Faster R-CNN models (e.g., stele area, average area of the late metaxylem).
 158 Furthermore, Faster R-CNN or Mask R-CNN models [54] can be trained to
 159 detect other objects, such as aerenchym and protoxylem objects, and their
 160 parameters, if data annotated with such objects becomes available.

161 *2.1. Overview of the Approach*

162 We have used Faster R-CNN [52], a state-of-the-art network for object
 163 detection, to detect objects of interest (i.e., root, stele, late metaxylem), and
 164 subsequently mark each object with a bounding box. More precisely, we
 165 have trained a Faster R-CNN model to identify the root and stele within a
 166 cross-section image, and another Faster R-CNN model to identify the late
 167 metaxylem within the stele region of a cross-section. Given the bounding
 168 box of an object, identified by the Faster R-CNN models trained on root
 169 cross-section images, we have calculated its diameter by averaging the width
 170 and height of the bounding box. The count of late metaxylem was obtained
 171 by counting the number of late metaxylem objects detected by the Faster
 172 R-CNN network.

173 The Faster R-CNN model architecture is shown in Figure 2. As can be seen,
 174 the Faster R-CNN has two main components. The first component consists
 175 of a Region Proposal Network (RPN), which identifies Regions of Interest

Figure 2: Faster R-CNN model architecture [52], which has two main components: 1) a region proposal network (RPN), which identifies regions that may contain objects of interest and their approximate location; and 2) a Fast R-CNN network, which classifies objects as root or stele, and refines their location, defined using bounding boxes. The two components share the convolutional layers of the pre-trained VGG-16 [55].

176 (i.e., regions that may contain objects of interest), and also their location.

177 The second component consists of a Fast R-CNN [56], which classifies the

178 identified regions (i.e., objects) into different classes (e.g., root and stele), and

179 also refines the location parameters to generate an accurate bounding box for

180 each detected object. The two components share the convolutional layers of

181 the VGG-16 network [55], which is used as the backbone of the Faster R-CNN

182 model. More details on convolutional neural networks, VGG-16 and Faster
183 R-CNN approach, which we used to detect objects and generate bounding
184 boxes, are provided below.

185 *2.1.1. Convolutional Neural Networks and VGG-16*

186 Convolutional Neural Networks (CNNs) [57] are widely used in image
187 analysis. While originally designed for image classification, the features
188 extracted by CNNs are informative for other image analysis tasks, including
189 object detection. A CNN consists of convolutional layers followed by non-
190 linear activations, pooling layers and fully connected layers, as seen in Figure
191 3 (which shows a specific CNN architecture called VGG-16 [55]).

192 A convolutional layer employs a sliding window approach to apply a set of
193 filters (low-dimensional tensors) to the input image. The convolution operation
194 captures local dependencies in the original image, and it produces a feature
195 map. Different filters produce different feature maps, consisting of different
196 features of the original image (e.g., edges, corners, etc.). A convolution layer
197 is generally followed by a non-linear activation function, such as the Rectified
198 Linear Unit (i.e., ReLU), applied element-wise to generate a rectified feature
199 map. The ReLU activation replaces all the negative pixels in a feature map
200 with zero values. A pooling layer is used to reduce the dimensionality of

Figure 3: VGG-16. The original VGG-16 architecture consists of 13 convolution+ReLU layers, five pooling layers, and three fully connected layers. A convolution+ReLU layer produces a feature map, while a pooling layer reduces the dimensionality of the feature map. The last fully connected layer uses a softmax activation function to predict one of the 1000 categories. The dimensions corresponding to each layer are also shown.

the rectified feature map. Intuitively, the pooling operation retains the most important information in a feature map by taking the maximum or the average pixel in each local neighborhood of the feature map. As a consequence, the feature map becomes equivariant to scale and translation [58].

After a sequence of convolutional layers (together with non-linear activations) and pooling layers, a CNN has one or more fully connected layers. In a fully connected layer, all neurons are connected to all neurons in the

subsequent layer. The first fully connected layer is connected to the last downsized feature map. The fully connected layers are used to further reduce the dimensionality and to capture non-linear dependencies between features [58]. The last fully connected layer uses a softmax activation function, and has as many output neurons as the number of targeted classes.

There are several pre-trained CNN architectures available, including VGG-16 [55], shown in Figure 3. The VGG-16 network has been shown to give very good performance in the ImageNet competition, where the network was trained on millions on images with 1000 categories [55]. Furthermore, VGG-16 was used with good results in the original Faster R-CNN study [52], which motivated us to use it also in our study. As can be seen in Figure 3, VGG-16 has 13 convolutional+ReLU layers, 5 pooling layers, and 3 fully connected layers. The dimensions corresponding to each layer are also shown in Figure 3.

2.1.2. Region Proposal Network (RPN)

As mentioned above, the region proposal network identifies regions that could potentially contain objects of interest, based on the last feature map of the pre-trained convolutional neural network that is part of the model, in our case VGG-16 [55]. More specifically, using a sliding window approach, k

regions are generated for each location in the feature map. These regions, are represented as boxes called *anchors*. The anchors are all centered in the middle of their corresponding sliding window, and differ in terms of scale and aspect ratio [52], to cover a wide variety of objects. The region proposal network is trained to classify an anchor (represented as a lower-dimensional vector) as containing an object of interest or not (i.e., it outputs an “objectness” score), and also to approximate the four coordinates of the object (a.k.a., location parameters). The ground truth used to train the model consists of bounding boxes provided by human annotators. If an anchor has high overlap with a ground truth bounding box, then it is likely that the anchor box includes an object of interest, and it is labeled as positive with respect to the *object* versus *no object* classification task. Similarly, if an anchor has small overlap with a ground truth bounding box, it is labeled as negative. Anchors that don’t have high or small overlap with a ground truth bounding box are not used to train the model. During training, the positive and negative anchors are passed as input to two fully connected layers corresponding to the classification of anchors as containing *object* or *no object*, and to the regression of location parameters (i.e., four bounding box coordinates), respectively. Corresponding to the k anchors from a location, the RPN network outputs $2k$ scores and $4k$

246 coordinates.

247 2.1.3. *Fast R-CNN*

248 Anchors for which the RPN network predicts high “objectness” scores
 249 are passed to the last two layers (corresponding to object classification and
 250 location parameter refinement, respectively) of a network that resembles
 251 the original Fast R-CNN network [56], except for how the proposed regions
 252 are generated. Specifically, in the original Fast R-CNN, the regions were
 253 generated from the original image using an external region proposal method
 254 (e.g., selective search).

255 As opposed to the original Fast R-CNN [56], in the Fast R-CNN component
 256 of the Faster R-CNN model, the external region proposal method is replaced by
 257 an internal RPN trained to identify regions of interest [52]. Highly overlapping
 258 regions, potentially corresponding to the same object, can be filtered using
 259 a non-maximum suppression (NMS) threshold. A pooling layer is used to
 260 extract feature vectors of fixed length for the regions of the interest proposed
 261 by RPN. Subsequently, the feature vectors are provided as input to two fully
 262 connected layers, corresponding to the classification of the object detected
 263 and the regression of its location, respectively.

264 The object classification layer in Fast R-CNN uses the softmax activation,

while the location regression layer uses linear regression over the coordinates defining the location as a bounding box. All parameters of the network are trained together using a multi-task loss [56].

2.1.4. *Faster R-CNN Implementation and Training*

The implementation of the original Faster R-CNN model [52], which is publicly available at https://github.com/ShaoqingRen/faster_rcnn, uses MATLAB as the programming language, and Caffe (<http://caffe.berkeleyvision.org>) as the backend deep learning framework. Chen and Gupta [59] provided an implementation of the Faster R-CNN model, which uses Python as the programming language and TensorFlow (<https://www.tensorflow.org>) as the backend deep learning framework. This implementation, publicly available at <https://github.com/endernewton/tf-faster-rcnn>, allows the user to train a model from scratch and also to reuse one of several pre-trained models as the backbone of the network. In particular, the user can select the VGG-16 network, pre-trained on the ImageNet dataset with 1000 categories.

We used the Python/TensorFlow implementation of the Faster R-CNN network, with the pre-trained VGG-16 model as its backbone, and trained the network to identify objects such as root, stele and late metaxylem. More precisely, the parameters of the VGG-16 convolutional layers, which are shared

284 by the Fast R-CNN and RPN networks in Faster R-CNN, were initialized
 285 using the pre-trained VGG-16 network. As many image features are highly
 286 transferable between different datasets, this initialization based on VGG-16
 287 allowed us to train accurate models from a relatively small number of root
 288 cross-section labeled images. In our preliminary experimentation, we found
 289 that it is difficult to accurately detect late metaxylem at the same time with
 290 root and stele. To address this issue, we trained a Faster R-CNN model to
 291 detect root and stele from background (i.e., everything else in the image),
 292 and another Faster R-CNN model to detect late metaxylem from background.
 293 To achieve this, we changed the output layer of the original Faster R-CNN
 294 network to reflect our classes (corresponding to the objects detected).

295 Given that the RPN and Fast R-CNN networks share 13 convolutional
 296 layers (initialized based on VGG-16), they were co-trained using an iterative
 297 process that alternates between fine-tuning the RPN and fine-tuning the Fast
 298 R-CNN network (with fixed proposed regions produced by RPN) [52]. All
 299 the model parameters were updated using stochastic gradient descent (SGD).

300 *2.2. Existing Approaches for Root Anatomy*

301 There are several approaches and tools for quantifying root anatomical
 302 variation based on cross-section images [47, 48, 49]. Approaches in this cate-

gory can be roughly categorized as manual, semi-automated, and automated approaches. Manual analysis of root images relies heavily on subjective assessments, and is suitable only for low throughput analysis. ImageJ [60] is an image analysis tool that has been extensively used to manually identify and quantify root anatomical traits [44, 61, 53], given that it enables researchers to mark objects of interest and obtain their measurements. In particular, the ImageJ software was used to acquire the ground truth (in terms of quantitative annotations) for the images used in this study, specifically, RD, SD, LMXD and LMXN measurements.

Semi-automated tools require user feedback to tune parameters for individual images in order to get accurate results. *RootScan* [47] and *PHIV-RootCell* [49] are semi-automated tools that identify and quantify anatomical root traits. RootScan was originally designed for analyzing maize root cross-section images. The analysis of each image involves several steps. RootScan starts by isolating the cross-section from the background using a global thresholding technique [62]. Subsequently, the stele is segmented based on the contrast between pixel intensities within and outside the stele. Different cells within the stele (e.g., late metaxylem) are classified based on their area according to background knowledge on root anatomy for a particular species. RootScan

322 can detect several types of objects (including lucunae, metaxylem and pro-
 323 toxylem), and also a broad range of parameters for each detected object. After
 324 each step, the user has to “approve” the automated detection or alternatively
 325 correct it, before moving to the next step. The tool can be run on a set of
 326 images in batch mode, but the user still needs to provide input for each step
 327 of the analysis for each image, as explained above.

328 The *PHIV-RootCell* tool for root anatomy is built using the ImageJ
 329 software [60], and provides options for selecting regions of interest (ROI) such
 330 as root, stele, xylem, and for measuring properties of these regions. It was
 331 designed for analyzing rice root cross-section images. Similar to RootScan,
 332 domain knowledge is used to identify ROIs. The PHIV-RootCell tool uploads
 333 and analyzes one image at a time, and does not have an option for batch
 334 uploading or processing. Furthermore, it requires user’s supervision at each
 335 segmentation and classification step [49]. For example, it requires the user
 336 to validate the root selection, stele selection, central metaxylem selection,
 337 among others.

338 As opposed to semi-automated tools that require user feedback, a fully
 339 automated approach should involve “a single click” and should produce
 340 accurate results without any human intervention during the testing and

341 evaluation phases. However, human input and supervision in the form of
 342 background knowledge or labeled training examples may be provided during
 343 the training phase. In this sense, *RootAnalyzer* [48] is an automated tool,
 344 which incorporates background knowledge about root anatomy. The first step
 345 in *RootAnalyzer* is aimed at performing image segmentation to distinguish
 346 between root pixels (corresponding to boundaries of individual root cells) and
 347 background pixels. To achieve this, *RootAnalyzer* utilizes a local thresholding
 348 technique to analyze each pixel's intensity by comparing it with the mean
 349 pixel intensity in a small square neighborhood around that pixel (defined by
 350 a width parameter, W). Subsequently, *RootAnalyzer* constructs a difference
 351 image, and classifies pixels as root or background pixels based on a threshold,
 352 T , used on the difference image. The next step is focused on detecting
 353 root cells and closing small leaks in cell boundaries, using an interpolation
 354 approach. Finally, cells are classified in different categories, such as stele cells,
 355 cortex cells, epidermal cells, etc. based on size, shape, and position. Two
 356 thresholds are used to classify cells as small or large: a threshold, A_s , for small
 357 cells, and a threshold, A_l , for large cells. Furthermore, stele cells are classified
 358 based on an additional threshold, N , on the maximum distance from a cell
 359 to any of its nearest neighbor cells. The *RootAnalyzer* tool can be used for

both single image processing and batch processing. Single image processing allows the user to adjust and tune parameters, and also to interact with the tool at each stage of the segmentation and classification. Batch processing requires the user to provide the parameters to be used with a specific batch of plant images. Similar to RootScan, RootAnalyzer outputs a table of area measurements and counts for regions of interest. This tool was designed for wheat and was shown to work also for maize [48].

2.3. Dataset

Twenty-five accessions of *Oryza* species were grown in plastic pots (25 cm in height; 26 and 20 cm diameter at the top and bottom, respectively), filled with 6 kg of clay loam soil. Three replications per each accession were maintained under well-watered conditions and roots were sampled 60 days after sowing, to ensure fully mature roots. The roots were harvested and washed thoroughly. To obtain the cross-section images used in this study, root samples stored in 40% alcohol were hand sectioned with a razor blade using a dissection microscope. For each of the 25 rice accessions, and for each of the three biological replicates, root samples from root-shoot junction and 6 cm from the root tip were obtained. Images of root sections were acquired with the Axioplan 2 compound microscope (Zeiss, Germany) at 50x and 100x

379 magnification. Specifically, for each accession and each replicate, 2-3 images
 380 were taken at root-shoot junction, and 2-3 images at 6 cm from the tip of the
 381 root, at 50x and 100x magnification. Thus, an image may have two versions:
 382 a 50 $\times$ magnification version, which captures the whole root diameter (top
 383 image in Figure 1), and a 100 $\times$ magnification version, which captures only the
 384 stele diameter (bottom image in Figure 1). However, not all 50 $\times$ images have
 385 a 100 $\times$ correspondent. Precisely, there are 388 images at 50 $\times$ magnification,
 386 and 339 images at 100 $\times$ magnification.

387 For each root image, we manually measured root anatomical parame-
 388 ters, such as root cross-section diameter, stele diameter, late metaxylem
 389 average diameter and late metaxylem number, using the ImageJ software
 390 [60]. Specifically, root diameters were estimated using the 50 $\times$ magnification
 391 images. The stele diameter, and late metaxylem average diameter and count
 392 were estimated using the 100 $\times$ magnification images, if available (otherwise,
 393 the 50 $\times$ magnification images were used). The manual measurements and
 394 counts constitute our ground truth to which we compared the measurements
 395 produced based on the bounding boxes detected by our trained Faster R-CNN
 396 models. Statistics about the dataset, including the minimum, maximum,
 397 average and standard deviation for RD, SD, LMXD and LMXN, are presented

in Table 1.

Statistics	RD	SD	LMXD	LMXN
Min	354	115	15	1
Max	1352	419	65	12
Avg $\pm$ std	869 $\pm$ 194	216 $\pm$ 55	36 $\pm$ 8	5.4 $\pm$ 1.8

Table 1: Ground Truth Statistics: minimum (Min), maximum (Max), and average together with standard deviation (Avg $\pm$ std) are shown for the ground truth measurements of RD, SD, LMXD (expressed in micrometers, μm) and LMXN (which is the count of late metaxylem objects).

In addition to measuring root anatomical parameters, each $50\times$ magnification image was also manually labeled by independent annotators with bounding boxes that represent root, stele, and late metaxylem, respectively, and each $100\times$ magnification image was labeled with boxes that represent late metaxylem.

We used the Labellmg tool available at <https://github.com/tzutalin/labelImg> to perform the bounding box labeling. This tool produces annotations in the Pascal Visual Object Classes (VOC) XML format [63], a standard format used for annotating images with rectangular bounding boxes corresponding to objects. An example of a root cross-section image annotated using the Labellmg tool is shown in Figure 4 (a), where each target object is marked using four coordinates, shown as green dots, which determine a bounding box. The bounding boxes annotated with the Labellmg tool in the $50\times$ and $100\times$

412 magnification images constitute the ground truth to which we compared the
 413 bounding boxes of the objects detected by our models. Corresponding to the
 414 ground truth image in Figure 4 (a) annotated with Labellmg, Figure 4 (b)
 415 shows the bounding box annotations produced by our models, as red boxes.

Figure 4: Objects of interests as bounding boxes: (a) Ground truth image annotated using Labellmg, where each object is marked using four coordinates, shown as green dots, which determine a bounding box. (b) The annotation of the same image by the root/stele and late metaxilem models, where the detected objects are shown using red bounding boxes.

416 We would like to emphasize that the $50\times$ magnification images contain
 417 all the anatomical features that we target in this study, and are sufficient
 418 for training the proposed deep learning models. However, we also trained
 419 models on the $100\times$ magnification images, independently, to understand how
 420 much the identification of the LMX objects and their measurements may be

improved by using images with a higher resolution. In general, any resolution can be used for training, as long as all the features that need to be identified are contained in the image.

2.4. Experimental Setup

2.4.1. Training, Development and Test Datasets

We performed a set of experiments using 5-fold cross-validation. Specifically, we split the set of $50\times$ magnification images into five folds, based on accessions, such that each fold contained 5 accessions out of the 25 accessions available. The exact number of $50\times$ magnification images (instances) in each fold is shown in Table 2. For each fold, Table 2 also shows the number of corresponding $100\times$ magnification images (instances) available (as mentioned before, not every $50\times$ magnification image has a corresponding $100\times$ magnification image). In each 5-fold cross-validation experiment, four folds were used for training, and the fifth fold was used for test. To tune hyper-parameters, we used one of the training folds as the development dataset. The results reported represent averages over the 5 folds. The reason for splitting the set of images based on accessions was to avoid using images from the same plant or the same replicate both in the training and test datasets.

Table 2: Number of instances in each of the 5 folds used to perform cross-validation for the 50 $\times$ and 100 $\times$ magnification images, respectively. The total number of instances in the dataset is also shown.

Fold	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Total
Instances (50 $\times$)	71	79	86	77	75	388
Instances (100 $\times$)	62	60	80	69	68	339

439 2.4.2. Evaluation Metrics

440 We used three standard metrics in our evaluation, driven by preliminary
441 observations. First, given that there exist exactly one root and one stele in
442 an image, we observed that these objects are always detected in the 50 $\times$
443 magnification images. We used the Intersection-over-Union (IoU) metric to
444 measure how well the predicted bounding boxes overlap with the ground
445 truth bounding boxes. Second, given that the number of LMX objects varies
446 between 1 and 12, and these objects are relatively small, the corresponding
447 object detection models are prone to both false positive and false negative
448 mistakes. Thus, we used mean average precision (mAP), a standard metric in
449 object detection, to evaluate the ability of our models to accurately identify
450 the LMX objects. Both IoU and mAP metrics range between 0 and 1, and
451 higher values are better. Finally, we used the root mean square error (RMSE)
452 and relative root mean square error (rRMSE) (i.e., percentage error) metrics
453 to measure the ability of the Faster R-CNN approach to detect objects

454 and corresponding bounding boxes that lead to root/stele/LMX diameter
455 measurements and LMX counts close to those available as ground truth. For
456 RMSE and rRMSE, smaller values are better.

457 *2.4.3. Hyper-parameter Tuning*

458 Deep learning models, in general, and the Faster R-CNN models, in
459 particular, have many tunable hyper-parameters. We tuned several hyper-
460 parameters shown to affect the performance of the Faster R-CNN models
461 [64], and used the values suggested by Ren et al. [52] for the other hyper-
462 parameters. More specifically, we tuned the IoU threshold used in the RPN
463 network to identify anchors that could potentially include an object of interest
464 (i.e., positive instances/anchors). Furthermore, we tuned the non-maximum
465 suppression (NMS) threshold which is used to filter region proposals produced
466 by the trained RPN network (specifically, if two proposals have IoU larger
467 than the NMS threshold, the two proposals will be considered to represent
468 the same object). At last, we tuned the fraction of positive instances in a
469 mini-batch.

470 The specific values that we used to tune the IoU threshold were 0.4, 0.5 and
471 0.6; the values used to tune the NMS threshold were 0.6, 0.7 and 0.8; and the
472 values used to tune the fraction of positive instances in a mini-batch were 1:5

473 and 1:4. To observe the variation of performance with the tuned parameters,
 474 and select the values that gave the best performance, we trained a model
 475 corresponding to a particular combination of parameters on three training
 476 folds, and evaluated the performance of the model on the development fold.
 477 The performance of the models for root and stele detection was measured
 478 using the IoU metric (by comparing the predicted bounding boxes with the
 479 ground truth bounding boxes), while the performance of the models for LMX
 480 detection was measured using the mAP metric (by comparing the detected
 481 LMX objects with the ground truth LMX objects) to ensure that the Faster
 482 R-CNN models can accurately detect all the LMX objects.

483 Our tuning process revealed that the performance did not vary signif-
 484 icantly with the parameters for our object detection tasks. However, the
 485 best combination of parameters for the root/stele models consisted of the
 486 following values: 0.4 for the IoU threshold, 0.8 for the NMS threshold and
 487 1:4 for the fraction of positive anchors in a mini-batch. The best combination
 488 of parameters for the LMX models was: 0.5 for the IoU threshold, 0.8 for the
 489 NMS threshold, and 1:4 for the fraction of positive anchors in a mini-batch.
 490 We used these combinations of values for the root/stele and LMX models,
 491 respectively, in our experiments described in the next section.

492 3. Results and Discussion

493 In this section, we present and discuss the results of our experiments
 494 using the Faster R-CNN models trained on rice root cross-section images.
 495 Furthermore, we outline time requirements for Faster R-CNN and discuss the
 496 availability of the Faster R-CNN model for root anatomy as a tool.

497 3.1. *Variation of Performance with the Number of Training Instances*

498 As opposed to the existing tools for identifying anatomical parameters in
 499 root cross-section images, which incorporate background knowledge about
 500 the root anatomy of a particular species and the types of images used, the
 501 automated Faster R-CNN approach is easily generalizable to various species
 502 and types of images, given that a representative set of annotated images
 503 is provided as training data. Under the assumption that data annotation
 504 is expensive and laborious, we aim to understand how many images are
 505 necessary for good performance on roots from a particular species. Intuitively,
 506 the number of required images should be relatively small, given that our
 507 model relies on a VGG-16 network pre-trained to detect a large number of
 508 objects, generally more complex than root, stele and late metaxylem objects.
 509 To validate our intuition, we have performed an experiment where we

510 varied the number of images used for training, while keeping the number of
 511 test images fixed. Specifically, we used 5, 10, 25, 50, 75, 100, 150, 200, 250,
 512 and all available training images in a split, respectively, to train models for
 513 detecting the root, stele and LMX in an image. The $50\times$ magnification images
 514 were used to train the models for root/stele/LMX. The $100\times$ magnification
 515 images were also used to train models for LMX, with the goal of understanding
 516 the benefits provided by higher resolution images. The trained models were
 517 subsequently used to detect root, stele, and LMX objects in test images.

518 The performance of the models was measured by comparing the predicted
 519 objects with the ground truth objects. We used the IoU metric to evaluate the
 520 predicted bounding boxes for root/stele by comparison with the corresponding
 521 ground truth bounding boxes. We used the mAP metric to measure the ability
 522 of the models to accurately detect LMX objects. The variation of performance
 523 with the number of training images is shown in Figure 5 for root/stele (Left
 524 plot) and LMX (Right plot). As can be seen, in the case of the models trained
 525 on the $50\times$ magnification images, the performance increases with the number
 526 of training images, but tends to stabilize generally around 250 images. This
 527 confirms our intuition that only a small number of labeled images is needed
 528 to learn accurate models for the problem at hand. Furthermore, the left

Figure 5: Variation of performance with the number of training images for root/stele detection model (Left plot), and for the LMX detection model (Right plot), respectively. We used 50 $\times$ magnification images to detect root and stele objects, and both 50 $\times$ and 100 $\times$ magnification images to detect LMX. The performance of the root/stele detection model was measured using the IoU metric (which shows how accurately the predicted bounding boxes match the ground truth), while the performance of the LMX detection model was measured using the mAP metric (which shows how accurately LMX objects were detected). The plots show average values over 5 splits together with standard deviation.

529 plot in the figure shows that the IoU values for both root and stele objects
530 are around 0.95, when all the training images are used, and that the root
531 bounding boxes are slightly better than the stele bounding boxes. Similarly,
532 the LMX objects are detected with high accuracy, as shown in the right plot
533 of Figure 5, where the mAP values are close to 0.9 consistently for models
534 trained with smaller or larger number of 100 $\times$ magnification images. Similar
535 performance is obtained with the models trained from all 50 $\times$ magnification
536 images. The plots for both root/stele and LMX also show that generally the
537 variance decreases with the size of the data. The slow decrease in performance

that is observed sometimes between two training set sizes can be explained by the addition of some inconsistently labeled images present in the original dataset. Examples of inconsistently labeled images as shown in Figure 6.

3.2. Performance Evaluation Using RMSE/rRMSE

The Faster R-CNN models trained on root images were used to detect root/stele/LMX objects in the test data. Subsequently, the detected objects were further used to calculate RD, SD, LMXD and LMXN. To evaluate the models in terms of their ability to produce accurate root/stele/LMX diameter and LMX number, we have used the RMSE error computed by comparing the measurement/count estimates obtained from the predicted bounding boxes with the ground truth measurements/counts. The RD and SD measurements were evaluated based on models trained/tested with the 50 $\times$ magnification images, while LMXD and LMXN were evaluated based on models trained/tested with 50 $\times$ and 100 $\times$ magnification images, respectively. Intuitively, the LMXD/LMXN results obtained with the models trained on the 100 $\times$ magnification images should be more accurate, as those images have higher resolution. The RMSE/rRMSE results of the experiments corresponding to the five splits, together with the average over the five splits, are shown in Table 3. In addition, Table 3 shows the expected human error estimated

Figure 6: Examples of inconsistent human annotations that are included in our ground truth dataset. Specifically, image (a) was manually labeled as having LMXN=4 (the smaller LMX was included in the count), while image (b) was labeled as having LMXN=3 (the smaller LMX was not included in the count although it has size comparable with the smaller LMX counted in (a)). Our models consistently identified 4 LMX objects in both (a) and (b) images. Similarly, image (c) was incorrectly labeled manually as having LMXN=4, while the similar image in (d) was properly labeled as having LMXN=3. Our models correctly identified 3 LMX objects in both (c) and (d) images. Finally, images (e) and (f) show a larger number of LMX which have variable size, but it is not very clear which LMX were counted by the human annotator and which were not counted to get the 12 and 11 counts, respectively. Our models identified 7 LMX objects in image (e) and 10 LMX objects in image (f).

557 by performing an additional manual annotation using ImageJ (similar to how
558 the original ground truth annotation was done), and comparing the second
559 manual annotation against the first manual annotation.

Table 3: RMSE (μm) and rRMSE (i.e., percentage error) results for root diameter (RD), stele diameter (SD), late metaxylem diameter (LMXD) and late metaxylem number (LMXN) for 5 splits, together with the average over the 5 splits, and also the estimates for the human error. The number of $50\times$ magnification images used in these experiments is 388, while the number of $100\times$ magnification images is 339. For each measurement, the magnification of the images used to train the model that produced that measurement (i.e., $50\times$ or $100\times$) is also shown.

Split	RD($50\times$)		SD($50\times$)		LMXD($50\times$)		LMXD($100\times$)		LMXN($50\times$)		LMXN ($100\times$)	
	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE
Split 1	62.77	6.78	21.93	9.16	3.67	9.50	2.45	6.54	0.81	22.34	1.37	24.55
Split 2	32.18	3.94	17.54	8.32	3.77	10.53	3.13	8.18	0.71	16.55	0.45	9.17
Split 3	61.19	6.90	21.96	9.16	3.53	9.07	3.22	7.87	0.91	17.35	0.83	15.53
Split 4	33.12	3.74	20.01	9.18	3.58	11.70	3.56	10.34	1.90	30.98	0.63	11.33
Split 5	43.67	3.26	20.94	10.26	2.43	7.51	1.61	4.61	0.74	16.39	0.25	5.02
Average	46.59	4.92	20.39	9.21	3.40	9.66	2.79	7.51	1.02	20.72	0.71	13.12
Human error	48.14	5.46	25.17	11.29	3.39	9.13	3.39	9.13	0.21	3.89	0.21	3.89

560 As can be seen from Table 3, the average RMSE error for RD over the
561 5 splits is $46.59\mu m$, while the average rRMSE is 4.92%. Given that root
562 diameter for the images in our dataset varies between $354\mu m$ and $1352\mu m$ (see
563 Table 1), and that the RMSE estimate for human error for RD is $48.14\mu m$
564 (with the corresponding rRMSE being 5.46%), these results suggest that
565 the Faster R-CNN models trained on rice images can accurately learn to
566 predict RD. Similarly, the average RMSE error for SD over the five splits is
567 $20.39\mu m$ and the corresponding rRMSE is 9.21%, while the stele diameter

varies between $115\mu m$ and $419\mu m$. As for RD, the RMSE/rRMSE errors for the SD predictions are smaller than the estimates for human error, which are $25.17\mu m$ and 11.29%, respectively. As opposed to root and stele, the LMXD is significantly smaller, varying between $15\mu m$ and $65\mu m$. In this case, the average RMSE error is $3.40\mu m$ and $2.79\mu m$ for models trained using $50\times$ and $100\times$ magnification images, respectively. The rRMSE for the model trained on the $50\times$ magnification images is 9.66%, and decreases to 7.51% for the model trained on the $100\times$ magnification images. Compared with the SD estimates for human error (which are based on the $100\times$ magnification images, when available, or the $50\times$ magnification images, otherwise), the results of the models trained on the $50\times$ magnification images are slightly worse (rRMSE is 9.66% versus 9.13%), while the results of the models trained on the $100\times$ magnification images are slightly better (7.51% versus 9.13%).

In terms of LMXN, the ground truth numbers vary between 1 and 12, with an average of 5 LMX objects per image. The average RMSE error for LMXN is 1.02 for models trained on $50\times$ magnification images and 0.71 for models trained on $100\times$ magnification images. Correspondingly, the rRMSE is 20.70% for models trained on $50\times$ magnification images, and down to 13.12% for models trained on $100\times$ magnification images. While the Faster R-CNN

models trained with the $100\times$ magnification images reduce the rRMSE error by approximately 7.5%, their average error is still higher than the estimate for human error by approximately 10%, showing that these models could be further improved with more training data.

We performed error analysis to gain insights into the usefulness of these results in practice. Specifically, we analyzed images where our models made mistakes in terms of LMXN, and observed that some of those images were annotated in an inconsistent way by the human annotators, as can be seen in Figure 6, where some smaller LMX objects are sometimes counted and other times not counted. This observation is not surprising, as human annotators are prone to mistakes and inconsistencies. As opposed to that, the automated Faster R-CNN models produce more consistent results (i.e., consistently count or not count a smaller LMX). More training images are necessary to learn well in the presence of noise/inconsistencies. Nevertheless, our results suggest that the Faster R-CNN approach to root anatomy has the potential to replace the labor-intensive manual annotations of root cross-section images.

3.3. *Faster R-CNN Robustness to Image Variations*

We further studied the ability of the Faster R-CNN models to “adapt” to other types of root cross-section images. To do this we identified 14 images

606 that have been used to demonstrate RootAnalyzer and 10 images that have
 607 been used to demonstrate PHIV-RootCell. In addition, we also searched the
 608 Web for root cross-section images, and identified 15 more images from rice,
 609 9 images from maize, and 9 images labeled as monocot root cross-section
 610 images. Together, our dataset of *external* images consists of 57 heterogeneous
 611 images, which came from different species, were taken with different imaging
 612 technologies under different conditions, had different sizes and resolutions,
 613 different backgrounds, different luminosity, etc. We randomly split each
 614 category of images into training/validation and test subsets. Specifically, 42
 615 images were used for training/validation and 15 images were used for test.
 616 We initially used the Split 1 models (trained on $50\times$ magnification images)
 617 to identify RD, SD, LMXD and LMXN traits for the external test images.
 618 Subsequently, we fine-tuned the Split 1 models with the external training
 619 images, and used the fine-tuned models to identify the RD, SD, LMXD and
 620 LMXN traits for the external test images. The results of these experiments
 621 are shown in Table 4.

622 As can be seen in the table, out-of-the-box, the Faster R-CNN models
 623 trained on our original rice images were not very accurate on the external
 624 images. In fact, the original models could not even detect the root in 4 out

Table 4: Faster R-CNN Model Robustness to Image Variations. The training and test internal images correspond to the training and test subsets of Split 1. The external images are collected from the Web. We used $\text{RMSE}(\mu\text{m})/\text{rRMSE}(\%)$ to compare models trained on internal images with models trained on internal and external images in terms of their ability to detect RD/SD/LMX objects (and derived their diameter) in a variety of images.

Experiment	RD (50%)		SD (50%)		LMXD (100%)		LMXN (100%)	
	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE
Train on internal images Test on external images	480.99	57.14	301.46	100.28	45.02	91.04	3.78	53.96
Train on internal/external images Test on external images	24.85	2.95	13.67	4.55	3.85	7.79	0.58	8.25
Train on internal images Test on internal images	62.77	6.78	21.93	9.14	3.67	9.50	0.81	22.34
Train on internal/external images Test on internal images	59.79	6.46	20.18	8.41	2.84	7.56	0.96	17.46

of 15 images, and could not detect the stele in 7 out of 15 images, due to the differences between the external images and our internal images used for training (if an object was not detected, a 0 diameter was assigned to it). However, the fine-tuned models significantly improved the results of the original models, with rRMSE dropping from 57.14% to 2.95% for RD, from 100.28% to 4.55% for SD, from 91.04% to 7.79% for LMXD, and from 53.96% to 8.25% for LMXN. We emphasize that the high errors of the original models are generally due to the models not being able to detect some objects at all (although the error for the objects detected was relatively small). These results show that the Faster R-CNN models fine-tuned with a small number of images (specifically, 42) can learn to predict the new types of images accurately.

637 To ensure that the performance of the fine-tuned models was not worse
 638 than the performance of the original models on our internal images, we also
 639 tested the fine-tuned models on the test fold corresponding to Split 1 (which
 640 was used for training). We recorded both the results of the original models and
 641 the results of the fine-tuned models in Table 4 (the last two rows, respectively).
 642 As can be seen, the results on our internal images improved slightly when
 643 using the fine-tuned models, as those models are more robust to variations.
 644 Specifically, rRMSE dropped from 6.78% to 6.46% for RD, from 9.14% to
 645 8.41% for SD, from 9.50% to 7.56% for LMXD, and from 22.34% to 17.46%
 646 for LMXN. It is also interesting to note that the results of the models on
 647 the external images are better than the overall results on the internal images.
 648 One possible reason for this may be that the images found online are generally
 649 clearer images, used to illustrate root anatomy, despite the fact that they are
 650 different from our internal images.

651 3.4. *Advantages and Disadvantages of the Faster R-CNN Approach*

652 While a direct comparison between the Faster R-CNN model (trained
 653 on rice root cross-section images) and existing approaches (e.g., RootScan
 654 and RootAnalyzer) is not possible, given that each approach is trained on
 655 different species, in this section, we first outline several advantages of the

656 Faster R-CNN model by comparison with existing models, and then emphasize
657 several disadvantages.

658 Regarding the advantages, the following points can be made:

- 659 (1) For an existing tool, it is hard to find parameters that are universally
660 good for a set of images. For example, for a given set of parameters,
661 the segmentation result from the RootAnalyzer in Figure 7 shows that
662 the parameters are appropriate for the left rice image (a) where the
663 LMX are reasonably well identified, but not appropriate for the right
664 rice image (b) where no LMX is identified. As opposed to that, our
665 experiments have shown that the performance of the Faster R-CNN
666 model does not vary much with hyper-parameters. Once a model is
667 properly trained, it performs accurately on a variety of images.
- 668 (2) Plant samples used for imaging are grown in different conditions, for
669 example in hydroponic (water based nutrient supply) or in soil, and root
670 cross-section images are collected using different techniques (e.g., hand
671 sectioning or sectioning using tools like vibratomes). Plant growing or
672 image acquisition differences lead to differences in image's color, contrast
673 and brightness. As opposed to other tools, the Faster R-CNN model
674 is not very sensitive to the light conditions or to the structure of the

Figure 7: RootAnalyzer Annotations: With the same set of parameters, in the left image the root border (red), stele border (yellow), endodermis (green) and late-metaxylem (purple) are detected reasonably well, while in the right image, only half of the stele border is detected. Given that the tool fails to properly detect the stele border, it also fails to detect the late metaxylem.

675 root cross-section images (including the epidermis thickness, epidermis
676 transparency, and distorted cross-sections), assuming the models are
677 trained with a variety of root cross-section images.

678 (3) Each existing tool is designed with certain image characteristics in mind,
679 and may not work on images that do not exhibit those characteristics.
680 For example, RootAnalyzer assumes a clear cell boundary and does
681 not work for images that contain a solid boundary where the cells are
682 not clearly identifiable. The Faster R-CNN models simply reflect the
683 broad characteristics of the images that they are trained on, instead of
684 being built with some characteristics in mind. No specific background

685 knowledge is provided, except for what is inferred automatically from
686 training images.

687 (4) Each tool is designed for a particular species, and incorporates back-
688 ground knowledge for that particular species. As different species may
689 have different root anatomy, a tool designed for a species may not work
690 for other species. For example, RootAnalyzer is designed to automati-
691 cally analyze maize and wheat root cross-section images, and “may work”
692 for other species [48]. However, the Faster R-CNN model can be easily
693 adapted to other species, assuming some annotated training images
694 from those species are provided. No other background knowledge is
695 required. Along the same lines, the Faster R-CNN model can be easily
696 adapted to images with different resolutions, assuming those images
697 include the features of interest.

698 While the Faster R-CNN model presents several advantages as compared
699 to existing approaches that incorporate background knowledge, it also has
700 several disadvantages, as outlined below:

- 701 • We found that smaller LMX objects are not detected by the Faster
702 R-CNN models, most probably due to inconsistencies in the training
703 data, as illustrated in Figure 6. To better handle noise and improve the

704 performance, more training data might be needed. Alternatively, more
705 consistent ground truth should be provided.

706 • While the bounding boxes which mark detected objects can produce
707 accurate results, they are not always perfectly enclosing the detected
708 object, as it can be seen in Figure 4. Thus, the diameter measurements
709 can be sometimes slightly biased, and could potentially be improved.

710 • The Faster R-CNN model can accurately detect objects and identify
711 traits such as diameter for the detected objects. The diameter can
712 be subsequently used to derive other traits such as the object area.
713 However, better area estimates could be potentially obtained with a
714 Mask R-CNN model, which has the ability to detect object boundaries
715 more precisely.

716 • The Faster R-CNN models can detect objects that can be marked with
717 a bounding box. For other types of objects (e.g., aerenchym objects),
718 Mask R-CNN models may be more appropriate.

719 *3.5. Faster R-CNN Approach as a Tool for Root Anatomy*

720 The Faster R-CNN model trained on our images can be used as a tool
721 from a terminal or through a web-based application, which is also mobile

722 friendly. The web-based application is available at <https://rootanatomy.org>.
 723 This site is linked to a GitHub repository that contains the source code, the
 724 pre-trained Faster R-CNN models and the ground truth data. The web-based
 725 application is user-friendly and does not require any programming skills. It
 726 can be run with one of our sample images displayed on the site, or with an
 727 image uploaded by the user.

728 *3.6. Time Requirements*

729 In terms of time/image requirements, our experiments have shown that
 730 accurate Faster R-CNN models can be trained from scratch with 150 to 250
 731 images. The average time for labeling an image with LabelImg is approxi-
 732 mately 2 minutes. The average time for training a model on an EC2 p2-xlarge
 733 instance available from Amazon Web Services (AWS) is approximately 10
 734 hours, and does not require any human intervention during that time. Once
 735 the model is trained, the average time to annotate a new image is less than one
 736 second (using an EC2 p2-xlarge instance). If using our webserver (hosted on a
 737 local machine), the running time for annotating a new image is approximately
 738 9 seconds, as this includes the time to setup the virtual environment, the
 739 time to retrieve the input image from the server, the time to perform the
 740 annotation, and the time to download the image to the user's browser. Given

741 these time requirements, assuming that a relatively large number of images
 742 need to be annotated for a biological study (on the order of thousands),
 743 the human time can be potentially reduced from days or weeks (the time
 744 would take to manually annotate all images) to hours (the time may take to
 745 manually label images for training) or minutes (the time for automatically
 746 annotating images with our tool).

747 Furthermore, the human time for labeling images for training could be
 748 dramatically reduced to less than an hour, if one is fine-tuning the Faster
 749 R-CNN model trained on our images as opposed to training a model from
 750 scratch.

751 4. Conclusions

752 In this paper, we trained Faster R-CNN models on rice root cross-section
 753 images and used the trained model to perform root anatomy. The Faster
 754 R-CNN approach to root anatomy is fully automated and does not need
 755 any background knowledge, except for the implicit knowledge in images
 756 that the model is trained on. More specifically, we trained Faster R-CNN
 757 models to detect root, stele and LMX objects, and to predict bounding boxes
 758 for each detected object. Subsequently, the bounding boxes were used to

759 obtain anatomical properties, specifically, root diameter, stele diameter, LMX
760 diameter and LMX number. The Faster R-CNN models used had VGG-16
761 as a backbone, to take advantage of the extensive training of the VGG-16
762 network, and were fine-tuned on root cross-section images.

763 We evaluated the Faster R-CNN models in terms of their ability to
764 detect the objects of interest, and also in terms of their ability to lead
765 to accurate measurements for RD, SD, LMXD and LMXN. The results of
766 the evaluation showed that the models produced accurate and consistent
767 annotations, when trained on a relatively small number of training images,
768 specifically close to 300 images. For LMXD and LMXN, we trained Faster
769 R-CNN models from both 50 $\times$ magnification images and 100 $\times$ magnification
770 images. Our results showed that the performance is slightly better for the
771 100 $\times$ magnification images, although this magnification is not a requirement
772 for good performance. Furthermore, our results suggest that the Faster R-
773 CNN models can potentially be used in practice to accelerate the speed at
774 which root cross-section images are analyzed, and save significant human
775 efforts and costs.

776 The evaluation in this paper was done on rice images. However, an
777 important observation was that the models can be easily adapted to other

778 types of root cross-section images and also to other species, by fine-tuning
 779 the existing Faster R-CNN models with a small number of labeled images
 780 from the species of interest. Similarly, additional anatomical features can be
 781 extracted by fine-tuning the existing models with images labeled according
 782 to other traits that are targeted (assuming the traits can be marked using
 783 bounding boxes).

784 While a direct comparison with existing tools for analyzing root cross-
 785 section images was not possible, we identified several advantages of the
 786 automated Faster R-CNN approach as compared to existing approaches that
 787 explicitly use background knowledge. We also identified several limitations
 788 of the Faster R-CNN model, including the fact that they can only be used
 789 for objects that can be represented using bounding boxes. As opposed to the
 790 Faster R-CNN model, existing approaches can identify a bigger variety of
 791 anatomical features. Thus, we can conclude that the Faster R-CNN approach
 792 and the existing tools have complementary strengths, and one cannot fully
 793 replace another.

794 As part of future work, we plan to thoroughly study domain adaptation
 795 approaches that allow the transfer of knowledge from the trained rice Faster
 796 R-CNN models to models for other plant species (or for other traits), without

797 labeling a large number of images from the other species of interest.

798 **Conflict of Interest Statement**

799 The authors declare that the research was conducted in the absence of any
800 commercial or financial relationships that could be construed as a potential
801 conflict of interest.

802 **Author Contributions**

803 XL carried out the original model adaptation and training, with input
804 from DC. CW, XL and DC carried out the computational experiment design,
805 with input from SVKJ. RB and SVKJ performed the biological experiment
806 design and collection of the data. CW and XL carried out the computational
807 experiments. RB performed the labeling of the data according to RD, SD,
808 LMXD and LMXN measurements. CW and XL performed the bounding box
809 labeling. SVKJ is the agronomy project leader with technical background in
810 root phenotyping. DC is the computational project leader, with background
811 in machine learning and deep learning. CW and XL drafted the first version
812 of the manuscript, and DC and SVKJ contributed to the preparation of
813 the final version of the manuscript. RJ contributed biological knowledge

814 to the manuscript and provided feedback on the preliminary version. CW
815 designed and developed the webserver. All authors read and approved the
816 final manuscript.

817 **Funding**

818 Contribution No. 19-072-J from Kansas Agriculture Experiment Station.

819 **Data Availability Statement**

820 The image datasets used in this study can be found in a GitHub repository
821 at <https://github.com/cwang16/Root-Anatomy-Using-Faster-RCNN>.

822 **Acknowledgments**

823 An earlier version of this manuscript has been released as a Pre-Print at
824 <https://www.biorxiv.org/content/10.1101/442244v2.article-info> [65].

825 **References**

826 [1] J. L. Araus, G. A. Slafer, C. Royo, M. D. Serret, Breeding for yield
827 potential and stress adaptation in cereals, *Critical Reviews in Plant*
828 *Sciences* 27 (6) (2008) 377–412.

- 829 [2] G. S. Khush, Strategies for increasing the yield potential of cereals: case
830 of rice as an example, *Plant Breeding* 132 (5) (2013) 433–436.
- 831 [3] A. Bishopp, J. Lynch, The hidden half of crop yields, *Nature Plants* 1
832 (2015) 15117.
- 833 [4] J. Jung, S. Mccouch, Getting to the roots of it: Genetic and hormonal
834 control of root architecture, *Frontiers in plant science* 4 (2013) 186.
- 835 [5] J. E. Schmidt, A. C. Gaudin, Toward an integrated root ideotype for
836 irrigated systems, *Trends in Plant Science* 22.
- 837 [6] Y. Uga, K. Sugimoto, S. Ogawa, J. Rane, M. Ishitani, N. Hara, Y. Kitomi,
838 Y. Inukai, K. Ono, N. Kanno, H. Inoue, H. Takehisa, R. Motoyama,
839 Y. Nagamura, J. wu, T. Matsumoto, T. Takai, K. Okuno, M. Yano,
840 Control of root system architecture by deeper rooting 1 increases rice
841 yield under drought conditions, *Nature genetics* 45.
- 842 [7] A. Bucksch, J. Burridge, L. M. York, A. Das, E. Nord, J. S. Weitz, J. P.
843 Lynch, Image-based high-throughput field phenotyping of crop roots,
844 *Plant Physiology*.
- 845 [8] T. Colombi, N. Kirchgessner, C. A. Le Marié, L. M. York, J. P. Lynch,

- 846 A. Hund, Next generation shovelomics: set up a tent and rest, *Plant and*
847 *Soil* 388 (1-2) (2015) 1–20.
- 848 [9] S. J. Mooney, T. P. Pridmore, J. Helliwell, M. J. Bennett, Developing
849 x-ray computed tomography to non-invasively image 3-d root systems
850 architecture in soil, *Plant and Soil* 352 (1) (2012) 1–22.
- 851 [10] H. Schulz, J. A. Postma, D. van Dusschoten, H. Scharf, S. Behnke, *Plant*
852 *root system analysis from mri images*, in: G. Csurka, M. Kraus, R. S.
853 Laramée, P. Richard, J. Braz (Eds.), *Computer Vision, Imaging and*
854 *Computer Graphics. Theory and Application*, Springer Berlin Heidelberg,
855 Berlin, Heidelberg, 2013, pp. 411–425.
- 856 [11] C. N. Topp, A. S. Iyer-Pascuzzi, J. T. Anderson, C.-R. Lee, P. R. Zurek,
857 O. Symonova, Y. Zheng, A. Bucksch, Y. Mileyko, T. Galkovskyi, B. T.
858 Moore, J. Harer, H. Edelsbrunner, T. Mitchell-Olds, J. S. Weitz, P. N.
859 Benfey, 3d phenotyping and quantitative trait locus mapping identify
860 core regions of the rice genome controlling root architecture, *Proceedings*
861 *of the National Academy of Sciences* 110 (18) (2013) E1695–E1704.
- 862 [12] D. van Dusschoten, R. Metzner, J. Kochs, J. A. Postma, D. Pflugfelder,

- 863 J. Buehler, U. Schurr, S. Jahnke, Quantitative 3d analysis of plant roots
864 growing in soil using magnetic resonance imaging, *Plant Physiology*.
- 865 [13] J. Pfeifer, N. Kirchgessner, T. Colombi, A. Walter, Rapid phenotyping
866 of crop root systems in undisturbed field soils using x-ray computed
867 tomography, *Plant Methods* 11:14.
- 868 [14] T. L. Durham Brooks, N. D. Miller, E. P. Spalding, Plasticity of ara-
869 bidopsis root gravitropism throughout a multidimensional condition
870 space quantified by automated image analysis, *Plant Physiology* 152 (1)
871 (2010) 206–216.
- 872 [15] R. T. Clark, R. B. MacCurdy, J. K. Jung, J. E. Shaff, S. R. McCouch,
873 D. J. Aneshansley, L. V. Kochian, Three-dimensional root phenotyping
874 with a novel imaging and software platform, *Plant Physiology* 156 (2)
875 (2011) 455–465.
- 876 [16] R. Sozzani, W. Busch, E. P. Spalding, P. N. Benfey, Advanced imaging
877 techniques for the study of plant growth and development, *Trends in*
878 *Plant Science* 19 (5) (2014) 304 – 310.
- 879 [17] M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives,
880 and prospects, *Science* 349 (6245) (2015) 255–260.

- 881 [18] S. Y. Rhee, M. Mutwil, Towards revealing the functions of all genes in
882 plants, *Trends in plant science* 19 (4) (2014) 212–221.
- 883 [19] W. Korani, J. P. Clevenger, Y. Chu, P. Ozias-Akins, Machine learning as
884 an effective method for identifying true single nucleotide polymorphisms
885 in polyploid plants, *The Plant Genome* 12 (1).
- 886 [20] R. Kaundal, R. Saini, P. X. Zhao, Combining machine learning and
887 homology-based approaches to accurately predict subcellular localization
888 in arabidopsis, *Plant physiology* 154 (1) (2010) 36–54.
- 889 [21] N. Heslot, H.-P. Yang, M. E. Sorrells, J.-L. Jannink, Genomic selection
890 in plant breeding: a comparison of models, *Crop Science* 52 (1) (2012)
891 146–160.
- 892 [22] U. Lee, S. Chang, G. A. Putra, H. Kim, D. H. Kim, An automated,
893 high-throughput plant phenotyping system using machine learning-based
894 plant segmentation and image analysis, *PloS one* 13 (4) (2018) e0196615.
- 895 [23] A. L. Maddison, A. Camargo-Rodriguez, I. M. Scott, C. M. Jones, D. M.
896 Elias, S. Hawkins, A. Massey, J. Clifton-Brown, N. P. McNamara, I. S.
897 Donnison, et al., Predicting future biomass yield in miscanthus using

- 898 the carbohydrate metabolic profile as a biomarker, *Gcb Bioenergy* 9 (7)
899 (2017) 1264–1278.
- 900 [24] W. Jones, K. Alasoo, D. Fishman, L. Parts, Computational biology:
901 deep learning, *Emerging Topics in Life Sciences* 1 (3) (2017) 257–274.
- 902 [25] F. Tardieu, L. Cabrera-Bosquet, T. Pridmore, M. Bennett, Plant phe-
903 nomics, from sensors to knowledge, *Current Biology* 27 (15) (2017)
904 R770–R783.
- 905 [26] A. Singh, B. Ganapathysubramanian, A. K. Singh, S. Sarkar, Machine
906 learning for high-throughput stress phenotyping in plants, *Trends in*
907 *Plant Science* 21 (2) (2016) 110 – 124.
- 908 [27] M. P. Pound, J. A. Atkinson, A. J. Townsend, M. H. Wilson, M. Griffiths,
909 A. S. Jackson, A. Bulat, G. Tzimiropoulos, D. M. Wells, E. H. Murchie,
910 T. P. Pridmore, A. P. French, Deep machine learning provides state-
911 of-the-art performance in image-based plant phenotyping, *GigaScience*
912 6 (10) (2017) 1–10.
- 913 [28] S. T. Namin, M. Esmailzadeh, M. Najafi, T. B. Brown, J. O. Borevitz,
914 Deep phenotyping: Deep learning for temporal phenotype/genotype
915 classification, *bioRxiv* (2017) 134205.

- 916 [29] S. Aich, I. Stavness, Leaf counting with deep convolutional and deconvolutional networks, arXiv preprint arXiv:1708.07570.
- 917
- 918 [30] A. Dobrescu, M. V. Giuffrida, S. A. Tsaftaris, Leveraging multiple datasets for deep leaf counting, in: ICCV Workshops 2017, 2017, pp. 2072–2079.
- 919
- 920
- 921 [31] J. R. Ubbens, I. Stavness, Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks, *Frontiers in plant science* 8 (2017) 1190.
- 922
- 923
- 924 [32] Z. Khan, V. Rahimi-Eichi, S. Haefele, T. Garnett, S. J. Miklavcic, Estimation of vegetation indices for high-throughput phenotyping of wheat using aerial imaging, *Plant methods* 14 (1) (2018) 20.
- 925
- 926
- 927 [33] A. Kamilaris, F. X. Prenafeta-Boldú, Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture* 147 (2018) 70–90.
- 928
- 929 [34] S. P. Mohanty, D. P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, *Frontiers in plant science* 7 (2016) 1419.
- 930
- 931 [35] B. M. Delory, M. Li, C. N. Topp, G. Lobet, archidart v3. 0: A new data

932 analysis pipeline allowing the topological analysis of plant root systems,
933 F1000Research 7.

934 [36] M. P. Pound, S. Fozard, M. T. Torres, B. G. Forde, A. P. French, Auto-
935 root: open-source software employing a novel image analysis approach to
936 support fully-automated plant phenotyping, *Plant methods* 13 (1) (2017)
937 12.

938 [37] I. Betegón-Putze, A. González, X. Sevillano, D. Blasco-Escámez, A. I.
939 Caño-Delgado, Myroot: A novel method and software for the semi-
940 automatic measurement of plant root length, *bioRxiv*.

941 [38] C. Reeb, J. Kaandorp, F. Jansson, N. Puillandre, J.-Y. Dubuisson,
942 R. Cornette, F. Jabbour, Y. Coudert, J. Patiño, J.-F. Flot, et al., Quan-
943 tification of complex modular architecture in plants, *New Phytologist*
944 218 (2) (2018) 859–872.

945 [39] V. R. Gowda, A. Henry, A. Yamauchi, H. Shashidhar, R. Serraj, Root
946 biology and genetic improvement for drought avoidance in rice, *Field*
947 *Crops Research* 122 (1) (2011) 1–13.

948 [40] A. Henry, A. J. Cal, T. C. Batoto, R. O. Torres, R. Serraj, Root attributes

- 949 affecting water uptake of rice (*Oryza sativa*) under drought, *Journal of*
 950 *experimental botany* 63 (13) (2012) 4751–4763.
- 951 [41] R. Richards, J. Passioura, A breeding program to reduce the diameter
 952 of the major xylem vessel in the seminal roots of wheat and its effect on
 953 grain yield in rain-fed environments, *Australian Journal of Agricultural*
 954 *Research* 40 (5) (1989) 943–950.
- 955 [42] L. Comas, S. Becker, V. M. V. Cruz, P. F. Byrne, D. A. Dierig, Root
 956 traits contributing to plant productivity under drought, *Frontiers in*
 957 *plant science* 4 (2013) 442.
- 958 [43] J. P. Lynch, J. G. Chimungu, K. M. Brown, Root anatomical phenes
 959 associated with water acquisition from drying soil: targets for crop
 960 improvement, *Journal of Experimental Botany* 65 (21) (2014) 6155–6166.
- 961 [44] N. Kadam, X. Yin, P. Bindraban, P. Struik, K. Jagadish, Does morpho-
 962 logical and anatomical plasticity during the vegetative stage make wheat
 963 more tolerant of water-deficit stress than rice?, *Plant physiology* (2015)
 964 pp–114.
- 965 [45] M. Rieger, P. Litvin, Root system hydraulic conductivity in species with

966 contrasting root anatomy, *Journal of experimental botany* 50 (331) (1999)
 967 201–209.

968 [46] A. P. Wasson, R. Richards, R. Chatrath, S. Misra, S. S. Prasad, G. Re-
 969 betzke, J. Kirkegaard, J. Christopher, M. Watt, Traits and selection
 970 strategies to improve root systems and water uptake in water-limited
 971 wheat crops, *Journal of experimental botany* 63 (9) (2012) 3485–3498.

972 [47] A. L. Burton, M. Williams, J. P. Lynch, K. M. Brown, Rootscan: software
 973 for high-throughput analysis of root anatomical traits, *Plant and Soil*
 974 357 (1-2) (2012) 189–203.

975 [48] J. Chopin, H. Laga, C. Y. Huang, S. Heuer, S. J. Miklavcic, Rootanalyzer:
 976 a cross-section image analysis tool for automated characterization of root
 977 cells and tissues, *PloS one* 10 (9) (2015) e0137655.

978 [49] M. Lartaud, C. Perin, B. Courtois, E. Thomas, S. Henry, M. Bettem-
 979 bourg, F. Divol, N. Lanau, F. Artus, C. Bureau, et al., Phiv-rootcell:
 980 a supervised image analysis tool for rice root anatomical parameter
 981 quantification, *Frontiers in plant science* 5 (2015) 790.

982 [50] H. C. Hall, A. Fakhrzadeh, C. L. Luengo Hendriks, U. Fischer, Precision
 983 automation of cell type classification and sub-cellular fluorescence quan-

984 tification from laser scanning confocal images, *Frontiers in plant science*
985 7 (2016) 119.

986 [51] M. Sankar, K. Nieminen, L. Ragni, I. Xenarios, C. S. Hardtke, Auto-
987 mated quantitative histology reveals vascular morphodynamics during
988 arabidopsis hypocotyl secondary growth, *Elife* 3.

989 [52] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time
990 object detection with region proposal networks, in: *Advances in neural*
991 *information processing systems*, 2015, pp. 91–99.

992 [53] N. Kadam, A. Tamilselvan, L. M. F. Lawas, C. Quinones, R. Bahuguna,
993 M. J. Thomson, M. Dingkuhn, R. Muthurajan, P. Struik, X. Yin, K. Ja-
994 gadish, Genetic control of plasticity in root morphology and anatomy of
995 rice in response to water-deficit, *Plant physiology* (2017) pp–00500.

996 [54] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: *Proceedings*
997 *of the IEEE international conference on computer vision*, 2017, pp.
998 2961–2969.

999 [55] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-
1000 scale image recognition, *arXiv preprint arXiv:1409.1556*.

- 1001 [56] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international
1002 conference on computer vision, 2015, pp. 1440–1448.
- 1003 [57] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hub-
1004 bard, L. D. Jackel, Backpropagation applied to handwritten zip code
1005 recognition, Neural computation 1 (4) (1989) 541–551.
- 1006 [58] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015)
1007 436.
- 1008 [59] X. Chen, A. Gupta, An implementation of faster RCNN with study for
1009 region sampling, CoRR abs/1702.02138. [arXiv:1702.02138](https://arxiv.org/abs/1702.02138).
- 1010 [60] C. A. Schneider, W. S. Rasband, K. W. Eliceiri, Nih image to imagej:
1011 25 years of image analysis, Nature methods 9 (7) (2012) 671.
- 1012 [61] T. Yamauchi, K. Watanabe, A. Fukazawa, H. Mori, F. Abe, K. Kawaguchi,
1013 A. Oyanagi, M. Nakazono, Ethylene and reactive oxygen species are
1014 involved in root aerenchyma formation and adaptation of wheat seedlings
1015 to oxygen-deficient conditions, Journal of experimental botany 65 (1)
1016 (2013) 261–273.

- 1017 [62] N. Otsu, A threshold selection method from gray-level histograms, IEEE
1018 transactions on systems, man, and cybernetics 9 (1) (1979) 62–66.
- 1019 [63] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn,
1020 A. Zisserman, The pascal visual object classes challenge: A retrospective,
1021 International Journal of Computer Vision 111 (1) (2015) 98–136.
- 1022 [64] L. Zhang, L. Lin, X. Liang, K. He, Is faster r-cnn doing well for pedestrian
1023 detection?, in: European Conference on Computer Vision, Springer, 2016,
1024 pp. 443–457.
- 1025 [65] C. Wang, X. Li, D. Caragea, R. Bheemanahalli, S. K. Jagadish, Root
1026 anatomy based on root cross-section image analysis with deep learning,
1027 Cold Spring Harbor Laboratory, 10.1101/442244. arXiv:[https://www.](https://www.biorxiv.org/content/early/2019/02/06/442244.full.pdf)
1028 [biorxiv.org/content/early/2019/02/06/442244.full.pdf](https://www.biorxiv.org/content/early/2019/02/06/442244.full.pdf).