

ProTractor: a lightweight ground imaging and analysis system for early-season field phenotyping

Nico Higgs¹, Blanche Leyeza¹, Jordan Ubbens¹, Josh Kocur¹, William van der Kamp¹, Theron Cory¹,
Christina Eynck², Sally Vail², Mark Eramian¹, Ian Stavness¹

¹Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada

firstname.lastname@usask.ca

²Saskatoon Research and Development Center
Agriculture and Agri-food Canada
Saskatoon, SK, Canada

Abstract

Acquiring high-resolution images in the field for image-based crop phenotyping is typically performed by complicated, custom built “pheno-mobiles.” In this paper, we demonstrate that large datasets of crop row images can be easily acquired with consumer cameras attached to a regular tractor. Localization and labeling of individual rows of plants are performed by a computer vision approach, rather than sophisticated real-time geolocation hardware on the tractor. We evaluate our approach for cropping rows of early-season plants from a *Brassica carinata* field trial where we achieve 100% recall and 99% precision. We also demonstrate a proof-of-concept plant counting method for our ProTractor system using an object detection network that achieves a mean average precision of 0.82 when detecting plants, and an R^2 of 0.89 when counting plants. The ProTractor design and software are open source to advance the collection of large outdoor plant phenotyping datasets with inexpensive and easy to use acquisition systems.

1. Introduction

Early-season plant evaluations play an important role in plant breeding field trials and also inform crop management decisions that producers make. Plant emergence, counting the number of plants that emerge from a given number of planted seeds, is an important phenotype associated with seedling vigor and also a co-variate for analysis of yield plots [49]. Early-vigor, the rate of emergence and vegetative biomass accumulation, is important for generating photosynthates and competing with weeds. These high-value phenotypes are also some of the most time consuming to capture manually, e.g. crawling or bending to accurately count very small plants, and often change rapidly

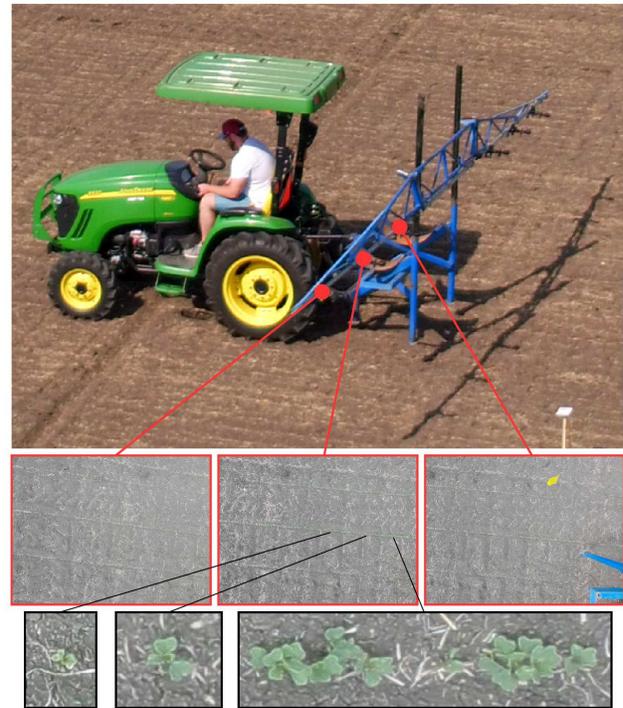


Figure 1: The *ProTractor* with images from the three left-side cameras. Cropped images (bottom) show three emergence patterns within the same row: a plant at the cotyledon stage (left), a plant with cotyledons and second true leaves (middle), and multiple clumped overlapping plants (right).

for short-season crops growing in long days, thus they are hard to evaluate at multiple time points. Therefore automated systems to estimate these phenotypes from field images are needed. This forms a challenging image acquisition and computer vision problem because the plants are small, rapidly growing, and often overlapping.

Ground-based platforms are commonly used for early season imaging because they provide higher resolution images than unmanned aerial vehicles (UAVs), which is important for small early-season plants. Ground platforms can also operate in windy conditions for which UAVs cannot fly, and therefore have a better chance of capturing images at the right stage of growth. Many ground-based platforms have been proposed [20, 8, 37, 10, 44]. Most are custom built vehicles that use custom instrumentation, including real-time geolocation, in order to associate images with specific locations within a field. Using consumer cameras on a commercial tractor would be a much simpler setup, less prone to equipment malfunction, and easier to replicate at multiple field locations. However, the simpler approach requires computer vision pre-processing to directly identify and label individual rows from raw image sequences without geolocation information. Early season phenotypes are often assessed on a row-by-row basis and require information from the entire row, e.g. the total number of plants within the row. Therefore, a ground-based platform that captures precisely geolocated images still requires a method to identify and crop out individual rows.

Plant counting from images has been previously studied in maize/sorghum [13, 50, 38] and soybean [48]. However, unlike large seeded crops like maize and soybean, seed singulation, consistent planting depth, and uniform spacing of seeds is difficult for smaller seeded crops, such as wheat and canola. This complicates emergence estimation because seeds often clump together resulting in clusters of overlapping plants which leads to sampling errors unless the entire row is assessed. Seeds at variable depth affect the consistency of emergence timing and subsequently, the growth stage of the plants within a row can vary. Emergence estimation in smaller seeded crops is also complicated by the fact that seeds are densely planted for higher target plant populations (25-35 plants/ft² for wheat, 5-10 plants/ft² for canola), as opposed to larger spacing within rows that is common for maize and sorghum plants (1 plant/ft² or less), with plants being much smaller size in general. Furthermore, seeding rates of canola are greater given low emergence rates (averages about 50-55%) and unpredictability of seedbed environments which greatly affect emergence [16].

Leaf phenotyping in rosette plants has been well studied, including previous leaf counting [11, 2] and segmentation [33, 36] competitions. These research advances were made possible because of publicly available rosette phenotyping datasets from controlled environments [29, 28]. However, translation of controlled environment phenotyping studies to field-based plant breeding has had limited success [3, 47], therefore, tackling image-based phenotyping directly from outdoor breeding trials is an important next step. The challenges of collecting field images are a barrier to the creation of public outdoor phenotyping datasets

similar to those that have been so successful for indoor rosette plants. Therefore we expect that a simple, inexpensive and fast field image capture system will help to spread outdoor field phenotyping. This will, in turn, help to advance the challenging computer vision problems associated with outdoor images, which are significantly more variable in their appearance due to cluttered backgrounds, overlapping plants, harsh and variable lighting conditions, and wind-induced plant motion.

In this paper, we present the *ProTractor*, a simple and inexpensive ground-based imaging platform, along with computer vision systems for pre-processing image sequences to extract and identify individual rows of plants in the field, and a deep-learning based method for emergence estimation in early-season Brassica plants (*B. carinata* and *B. napus* or canola). Our study makes the following contributions to computer vision in plant phenotyping:

1. a *field-to-phenotype* plant counting system with an open hardware design and open source software;
2. a new dataset of outdoor *B. carinata* images with individual plants annotated with bounding boxes, along with an analysis of inter-rater agreement;
3. a description of a lightweight and inexpensive ground imaging platform that can be easily replicated;
4. a general image analysis pipeline for identifying and cropping individual rows of plants from a sequence of top-down images, and its evaluation; and
5. an evaluation of an object detection architecture for plant detection/counting from early-season images.

2. Related Work

2.1. Ground-imaging platforms

Many robotic and human-operated systems for ground-based imaging in the field have been proposed for agronomic research. The systems vary in their sensing capabilities and level of automation, but are all created with the goal of collecting sensor data in outdoor field trials of plants. These systems encompass a wide variety of imaging strategies, depending on the specific target application. One common form factor for in-field imaging is the mobile push-cart or self-propelled vehicle with downward-facing sensors which images one or more rows at a time [20, 3]. These platforms have the advantage of flexibility, with a modular payload which can be replaced between imaging sessions. Also used are smaller autonomous robots which are capable of imaging individual plants in a row with a higher level of detail [46]. This is due to the higher proximity to the vehicle, as well as specialized sensors which are capable of performing tasks such as 3-D scanning and reconstruction. Others have utilized an in-place gantry system with a movable payload box carrying the sensor array [45]. This form factor has the advantage of not requiring

a vehicle to move through the field, a feature most commonly associated with remote sensing platforms such as UAVs. In contrast to vehicle-based platforms, this system trades off some of the flexibility and cost effectiveness of other systems. Also described is a crane equipped with a portable imaging booth which is capable of sheltering the plants from sunlight during imaging [41]. By isolating the plants in an imaging booth, the authors are able to illuminate the scene with light in specific wavelengths for multi-spectral imaging in the field. Like our proposed system, this system is able to be mounted on an existing tractor.

All of the field-based imaging systems described in the literature thus far are custom fabricated solutions, using purpose-built hardware. In addition, many rely on specialized sensors such as LIDAR or laser range finders [40, 46]. In contrast to these systems, the proposed system is easily replicated using commonly available consumer-grade cameras, and the agricultural equipment likely already present on site in order to manage the field trial. This makes the proposed system a feasible candidate for almost any field location, including in circumstances where deployment cost is a factor such as in developing countries.

2.2. Row detection

When using ground-based imaging techniques, individual images may contain one or more rows depending on the canopy distance as well as the camera’s field of view. Identifying individual rows of plants can be accomplished using a row centreline detection algorithm to find a line which best fits the centre of each row in the image. Several image processing techniques exist for row centreline detection, often for the purpose of enabling autonomous navigation of robotics in field environments. One such method segments the vegetation pixels using a vegetation index and utilizes a pixel membership heuristic combined with a horizontal scanline [51]. Other proposed techniques use the Hough Transform [18] or its variants to find points in the parameter space of all lines in the image which most correlate with vegetation pixels in the image space [19, 24]. However, such techniques are only applicable to rows which appear as straight lines in the image. For this reason, Vidovi et al. proposed a centreline detection algorithm based on energy minimization which is capable of detecting the centreline in curved rows [43]. Another common approach to row centreline detection is vegetation segmentation followed by the use of linear regression in the image space [30, 14]. Other techniques take advantage of the periodic property of parallel rows as they appear in images, utilizing a band-pass filter in order to detect the presence of rows at a consistent spacing [39]. This same approach removes the need for vegetation segmentation by imaging in the near-infrared wavelengths where plants appear with higher intensity than in RGB images.

2.3. Plant detection

The counting of seedlings in outdoor imaging has been explored in previous work, although most often from aerial images [13, 3, 21, 35]. Both image processing and deep learning methods for object detection or density estimation have been used for this purpose. Jin et al. proposed a supervised classification system which uses features extracted from human-labelled connected components of the thresholded image with a Support Vector Machine (SVM) [21], while other methods use engineered features with a feed-forward neural network [7]. In [27], the authors use a skeletonization method with engineered region features to relate connected components to plant count using a simple equation modeling the relationship. More recent methods have focused on the use of deep learning methods. In [35], wheat plant count is obtained by a convolutional neural network (CNN) through global regression from an image of a row to a single regression output. Similarly, [1] first generates a density map from regions in the image using a CNN and then reports the mean value over this map.

3. Methods

In this section, we describe the three main steps in the ProTractor software pipeline: image acquisition, row cropping and plant detection. The code is open source and available at <http://github.com/p2irc/protractor>.

3.1. Image Acquisition

Our goal was to create an inexpensive system to capture close-up top-down images of an entire field trial within a short period of time. This was achieved by mounting consumer-grade cameras to a repurposed sprayer assembly attached to a tractor (Figure 1). For our setup, we used a John Deere 2025r tractor and a sprayer assembly with 20-foot long booms extending to both sides. The sprayer assembly was attached using a three-point hitch, which allowed it to be used interchangeably with other tractors, while still providing a reasonably stable connection to the tractor. In practice, the sprayer booms did not sway or jitter by an appreciable amount and minimal motion blur was observed in the images. We used GoPro Hero6 cameras operating in Timelapse Photo mode with a frame rate of two images per second. Our flexible setup would apply well to other mounting arrangements and camera types.

The ProTractor was deployed daily in large field experiments of *B. carinata* and *B. napus* (canola). These trials include single or double row plots, grouped into large ranges, and therefore obtaining per-row images was critical. Rows were 3.2 m long, with 1.8 m spacing between ranges, which permitted the tractor to drive between ranges perpendicular to the rows of plants. We deployed three cameras on either side of the tractor, which extended over the entire row

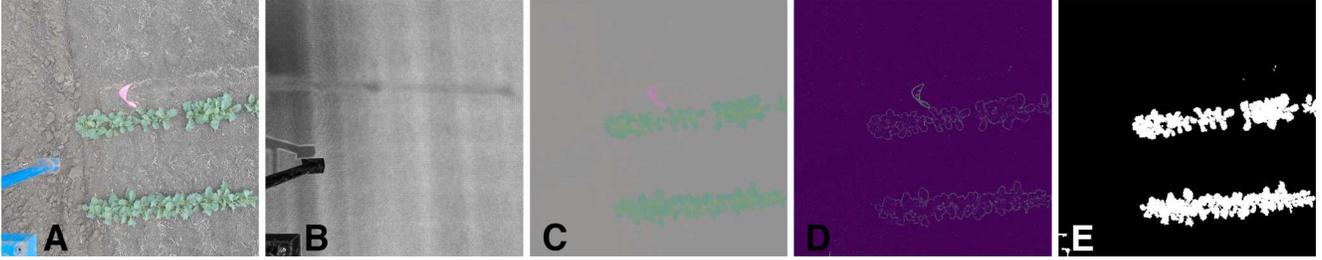


Figure 2: Pre-processing steps for binary segmentation of a raw ProTractor image (A). To mask away static portions of the image (parts of the sprayer boom and shadows), we sum over the change in pixels across all images to compute pixel gradients (B). To improve Otsu thresholding, we sample around plant edges found by flattening luminance (C) and forming gradient magnitudes (D). The excess green image is thresholded to generate a binary segmentation of plant pixels (E).

thanks to the sprayer booms. The goal was to capture side-by-side images that covered the entire row of plants, and then a sequence of images that covered the entire range as the tractor moved (see Figure 1, bottom). The side-by-side images were captured approximately synchronously by initiating all cameras at the same time with a wireless GoPro Remote. This permitted horizontal stitching to get an image of the entire row. For this study, the cameras were high enough that the centre camera captured the entire length of the row and could be used directly. To reduce the effect of shadows from the tractor/sprayer, images were captured with the tractor heading north (rows were seeded east-west).

Acquiring images with the ProTractor is quick and easy, but results in a set of images that pose several computer vision challenges such as: localizing images within the field without precise geolocation information, mitigating static objects that obstruct row identification, localizing multiple rows per image, differentiating sparse rows from soil and weeds, and identifying unique rows from overlapping images. These challenges are addressed by our software pipeline for cropping rows as described in the next section.

3.2. Row Cropping

The software pipeline to crop individual plant rows from a sequence of images is organized into three main steps. First, image pre-processing is used to identify pixels associated with plant matter. Second, rows are localized by fitting regression lines to the pixels associated with plants via our k-Lines algorithm. Third, rows are cropped and labelled using keypoint matching to identify similar rows in successive images.

3.2.1 Image Pre-Processing

Pre-processing was carried out on the ProTractor-captured image set in order to convert the raw RGB GoPro images into binary plant vs. non-plant (soil, crop residue, etc.) images. The pre-processing pipeline relies on thresholding the *excess green* ($ExG = 2G - R - B$) index with Otsu's

method [32] as illustrated in Figure 2. The resulting binary image is then passed as input to k-Lines for row cropping.

The excess green index is commonly used to isolate plant matter in images [15] and we found empirically that it provided a sufficient contrast between plant and non-plant pixels for our early-season *B. carinata* dataset. However, in order for excess green to be effective, problematic static portions of the images needed to be removed and the plant-to-non-plant ratio needed to be increased. Given the variation in outdoor lighting conditions while imaging the field, we also needed to automatically determine an optimal threshold value for each range of images, as opposed to a hand-tuned fixed value. The rest of this section describes these key pre-processing steps in greater detail.

Mounting cameras to an existing tractor, rather than a purpose-built vehicle, will likely result in static objects, such as the sprayer boom and associated shadows, appearing within the cameras' view. We mask these static object regions by iterating pairwise through the image sequence and recording the magnitude of the difference between each corresponding pair of RGB pixels. The sum of these differences over all image pairs is a single-band image β (Figure 2B). We then apply a custom threshold $t = 0.1 \times \max(\beta)$ to create a binary mask, with which we mask away static object regions from each image in the set.

Following the static object mask, we apply Otsu's method to each excess green image. Since our dataset was captured in the early season, the plant pixel count is very small relative to the non-plant pixel count. Additionally, the presence of crop residue and non-uniform texture in the soil adds significant noise to each image. In general, Otsu's method performs worse when object (plant) size is less than 30% of the background size, and when the signal-to-noise ratio is low [23]. We fix the low plant pixel count by creating a mask of the excess green image that has a more even ratio of plant to non-plant pixels. This is accomplished by sampling the luminance-flattened gradient magnitude image. We convert the image to the LAB colourspace and set each pixel's L value to the median L value of the en-

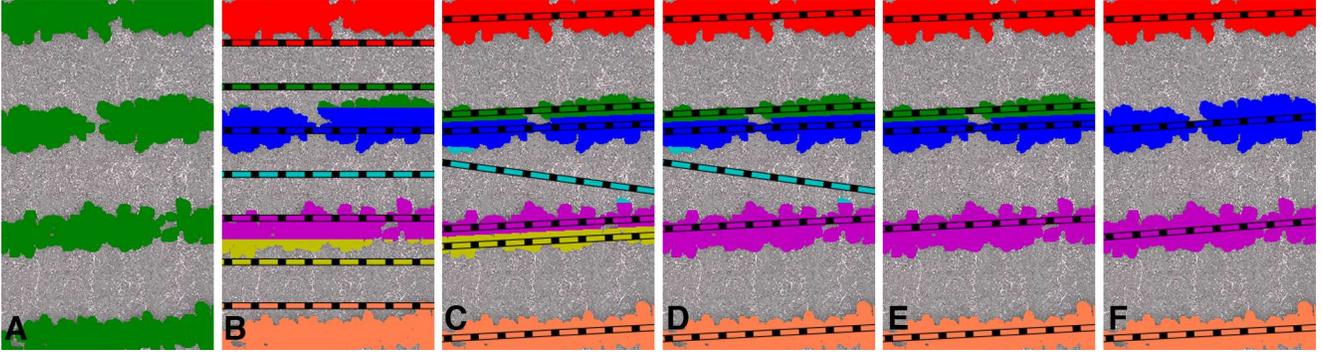


Figure 3: Steps in the k-Lines algorithm for row detection from the binary segmentation of plant pixels (A). Clusters are initialized with evenly-spaced horizontal lines (B). Lines are re-fit to clusters (C) and then we discard lines that are close together (yellow line, D) or intersecting (cyan line, E). The process iterates until convergence (F).

tire image sequence. We then convert the image back to RGB colourspace (Figure 2C). This luminance-flattening removes all edges in the image except for the edges that arise due to colour differences, which are mostly edges at the threshold between plant and non-plant matter. We locate these edges by computing the gradient magnitude of the image (Figure 2D). We then use the 50% strongest edges to create an edge mask and dilate the mask such that it encapsulates more than just the plants. The mask is applied to the excess green image to obtain an approximately equal distribution of plant vs. non-plant pixels and a higher signal-to-noise ratio. This masked excess green image serves as input for Otsu’s method.

In order to derive a binary threshold using Otsu’s method, we first compute separate threshold values for each individual excess green image. We then take the median of these thresholds and enforce it to be the minimum threshold for the entire data set. This is done because some images contain no plants and therefore cannot be thresholded using Otsu’s method. We apply the final threshold value to each excess green image (Figure 2E), and the resulting sequence of binary images serve as input to the k-Lines algorithm.

3.2.2 Identifying Rows with k-Lines

We identify rows within images with a clustering approach where we find groupings for plant pixels that ultimately represent each row, as outlined visually in Figure 3. Our k-Lines algorithm is similar to a general k-Models algorithm [4] which is an adaptation of the well-known k-Means algorithm. In k-Means, cluster centres are individual points and updated at each iteration to the mean value of the points in each cluster. Traditional k-Means struggles with our clustering problem for two main reasons. First, the distance metric greatly affects the shape of clusters; for example, standard Euclidean distance creates circular clusters opposed to clusters shaped like straight lines. While dif-

ferent distance functions can be used to change the shapes of clusters found, such as ignoring the horizontal axis in Euclidean distance to create horizontal clusters, it is not obvious if there is a distance function that can cluster lines of varying slope. Second, the resulting clusters are highly dependent upon the choice of k . While methods do exist to try to determine an appropriate k , such as finding the elbow in a scree plot, our problem requires a way to determine k more consistently and much more precisely than what is traditionally used in unsupervised clustering tasks.

In contrast to traditional k-Means clustering, our k-Lines algorithm re-imagines cluster centres as lines instead of points, and updates centres by refitting an ordinary least squares (OLS) regression line to the points associated with each “centre”. Because the cluster-centre update step and the cluster-assignment step are both minimizing the same function (vertical distance from the fitted line) in the same 2-dimensional Euclidean space, k-Lines retains the same convergence properties of k-Means. By using lines as centres and fitting them with an OLS regression we overcome the challenge of finding clusters that are in the shape of lines and our clusters can adapt to varying slopes.

Our k-Lines algorithm deals with the challenge of selecting a value for k by implementing heuristics for detecting and removing intersecting lines and lines that are deemed too close to one another. These heuristics arise naturally as constraints of our clustering task. Because the rows of plants are parallel to one another and a consistent distance apart, we can restrict our clustering task to only consider lines that do not intersect and that are at a specified minimum distance apart. The minimum distance between lines is derived from the following equations:

$$F_m = 2 \tan\left(\frac{F_\theta}{2}\right) h \quad (1)$$

$$\text{pixels per meter} = \frac{F_p}{F_m} \quad (2)$$

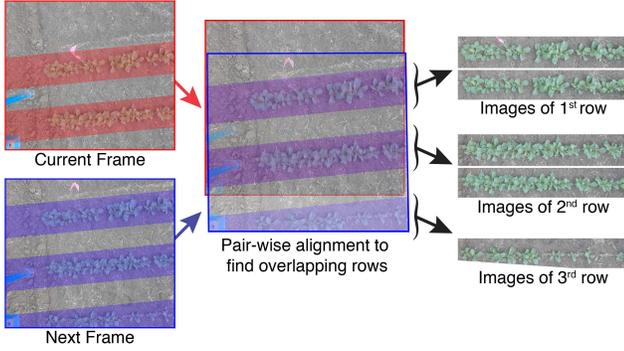


Figure 4: Our procedure for labeling unique rows aligns sequential pairs of images and groups overlapping rows under the same row label. The process is repeated for the entire image set for each range in the field.

$$\text{pixels per gap} = \text{pixels per meter} \times G_m \quad (3)$$

where F_θ is the camera’s vertical field of view in radians, F_m is the vertical field of view in meters along the ground, F_p is the vertical field of view in pixels (the height of the image in pixels), h is the height of the cameras above ground, and G_m is the gap between rows which is fixed based on the particular seeder used in the trial (1-foot row spacing is common). With the expected *pixels per gap* computed we take the minimum distance to be $0.75 \times \text{pixels per gap}$. With these heuristics, we can use a large initial value for k and the algorithm will reduce the number of lines by eliminating too close and intersecting lines until convergence.

3.2.3 Labelling and Cropping Identified Rows

Once rows have been localized with k-Lines, we need to determine their row number relative to the range they are in so that we can label them correctly when we crop them into individual row images. As described in Section 3.1, the ProTractor captures multiple images per second which results in images having considerable overlap. This means we have many cropped rows that are duplicates, which causes the labelling of rows to be nontrivial. To overcome this we create rectangles around the identified rows and use keypoint matching to map them to the previously processed image and detect duplicates by computing the intersection over union with the rows from the previous image (see Figure 4). Matched rows are given the same row label, and unmatched rows are considered new and given the incremented row label. For keypoint matching we use the OpenCV implementation [6] of the SURF algorithm [5] with fast approximate k-nearest neighbours matching, known as FLANN-based matching [31] to match images. With the large amount of overlap and similarity in images we found that more advanced techniques that generate more keypoints than SURF,

such as SIFT, showed no improvement in matching images and were in fact noticeably slower at runtime. However, simpler keypoint matching techniques that were fast, such as ORB, did show a noticeable decrease in their ability to match images. Once the correct row labels have been determined, the rows are rotated to a horizontal orientation based on the regression line fit from k-Lines, and then cropped to a standardized size for ease of use with computer vision and machine learning algorithms (see Figure 4).

3.3. Plant Detection

As a proof-of-concept for extracting early-season phenotypes with the ProTractor, we developed a plant detection method for row-by-row counting of plants in order to estimate the emergence rates for different lines of *B. carinata*. The plant detection problem here is posed similar to general object-detection tasks, using bounding box annotations on the row cropped images.

3.3.1 Data Annotation

We annotated plant row images obtained from the row cropper using LabelImg [42]. Six annotators created bounding boxes around the plants and the (x, y) centres and extent (widths, heights) were recorded. Annotators were instructed to look for the characteristic shape of two opposing cotyledons with possibly two “true leaves” growing perpendicularly from the “centre” of the plant when viewed from above (e.g., see example images in Figure 1), to draw bounding boxes as tightly around the plant as possible, and to permit boxes to overlap when plants overlap.

3.3.2 Inter-rater Agreement

We conducted an inter-rater reliability analysis to quantify the degree of agreement between raters. Six row images were annotated by seven raters using the criteria described above. None of the raters had extensive experience with the morphology of *B. carinata* or experience with counting *B. carinata* in overhead images. Of the six images, two were selected that had well-spaced-out plants, two with moderate amounts of plant overlap, and two with severe plant overlap.

The inter-rater agreement for the total number of plants identified in an image by annotators was tested by computing Krippendorff’s alpha [22], a non-parametric method with no restrictions on the number of raters, number of samples, or type of rating scale that also adjusts itself for small numbers of samples. This measure, on the scale from 0.0 to 1.0, was 0.95 indicating excellent agreement.

We also assessed the rate at which the majority of raters agreed upon the existence of a plant at a particular location. We used Density-based Spatial Clustering (DBSCAN) [12] to cluster bounding box centres into groups with at least 4 samples (a majority of the 7 raters) within a spatial radius of

5 pixels (Euclidean distance). The percentage of bounding box centres in all six images belonging to such a cluster, and deemed in agreement with the majority, was 82.9%.

Finally, for each such cluster of bounding boxes we computed the Generalized Tanimoto Coefficient (GTC) [9] for the extent of the bounding boxes, defined as

$$\text{GTC} = \frac{\sum_k \text{area}(R_k \cap S_k)}{\sum_k \text{area}(R_k \cup S_k)}$$

where (R_k, S_k) is the k -th pair of bounding boxes in the cluster. On a scale between 0.0 and 1.0, the average GTC of all such clusters of bounding boxes was 0.76, indicating moderately good agreement.

3.3.3 Faster R-CNN

For plant detection we used the Faster R-CNN [34] object detection system with a Feature Pyramid Network (FPN) [25] as the feature extractor and the ResNet-50 [17] architecture for the network backbone. The network was pre-trained on the COCO dataset [26], and then the head layers (FPN, RPN, and classifier and bounding box regressor layers) were retrained for the plant detection task. The network was trained for 375 epochs with a learning rate of $1e-4$. The Intersection-over-Union (IoU) threshold for the RPN was varied between 0.4 and 0.8.

To handle the high-resolution (4000×500) plant row images from the ProTractor, training and inference are performed on 200×200 patches taken from the full image, using a sliding window along the centreline of the row with 75% overlap for a total of 80 overlapping patches per row image. Out of 64 annotated row images, 52 were used as a training set and 6 each were used as validation and test sets. We applied standard data augmentation (image rotation, flipping) to the training set. At inference time, since the image patches overlap, we exclude bounding boxes for which 20% or more of the box is outside of the patch.

4. Results

We evaluated the proposed system on *B. carinata* field trials conducted by Agriculture and Agri-Food Canada. The layout of the field was 3 blocks \times 50 ranges \times 42 rows, for a total of 6300 unique rows within the trial. The row cropping software was tested on 44 ranges totalling 2018 of the 6300 unique rows from a single early-season date. First, we verified the ability of the row cropping algorithm to accurately label individual rows with the correct identifying number. Next, we evaluated an example phenotyping application where plant counting is performed on the row images using the proposed plant detection method.

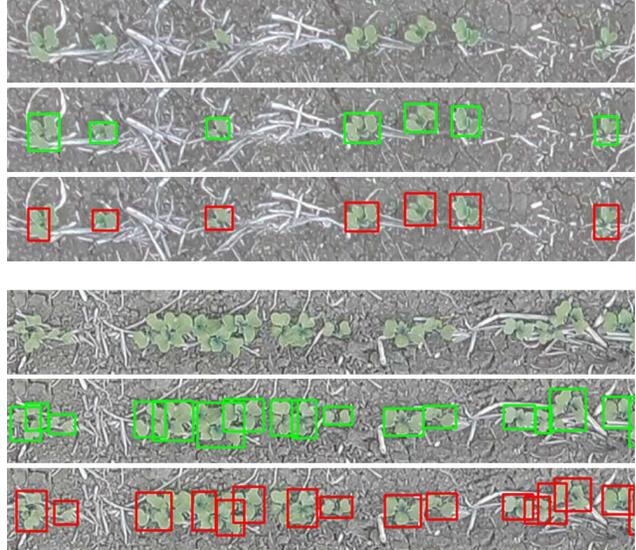


Figure 5: Annotations (green boxes) and predictions (red boxes) for example images with isolated plants (top) and with clumped overlapping plants (bottom).

4.1. Cropping of Rows

We evaluated the performance of the row cropping algorithm by determining the number of rows which are not correctly identified as rows, as well as the number of other objects in the image incorrectly identified as rows. This resulted in a precision of 0.99 and recall of 1.0. Row labels were generated by incrementing a counter as we traced through the image sequence. These labels were evaluated by comparing them with the known row labels from the field map. This resulted in a labeling accuracy of 0.80.

4.2. Plant Counting

For plant detection, we use mean average precision (mAP) to evaluate the quality of the bounding boxes output by Faster R-CNN (Table 1). Peak mAP was 0.82 at an IoU threshold of 0.4. Figure 5 shows examples of the bounding boxes output by the object detection model for cases of isolated plants and clumped plants. The final plant count for a given row is determined by the total number of bounding boxes predicted for that row. To evaluate the accuracy of the predicted plant count, we report mean difference in count (DiC), mean absolute difference in count ($|DiC|$), mean squared difference in count (MSE), and the coefficient of determination (R^2). As a baseline for comparison, we thresholded the images by excess green and performed a linear regression on the number of vegetation pixels. The results for both methods are shown in Table 2 and the distribution of count errors for Faster R-CNN is given in Table 3.

<i>IoU</i>	0.4	0.5	0.6	0.7	0.8
<i>mAP</i>	0.82	0.82	0.75	0.61	0.35

Table 1: Plant detection results for different *IoU* thresholds for the region proposal network.

	<i>DiC</i>	$ DiC $	<i>MSE</i>	R^2
Faster RCNN [34]	-1.71	2.57	10.86	0.89
Linear regression	-2.30	3.24	18.45	0.81

Table 2: Results for plant counting by object detection (Faster R-CNN) and by linear regression.

<i>DiC</i>	Number of Patches	% of Patches
-3	14	2.9%
-2	30	6.3%
-1	165	34.4%
0	244	50.9%
1	95	19.8%

Table 3: The distribution of Difference in Counts (*DiC*) for plant counting by object detection with Faster R-CNN.

5. Discussion

In this paper, we have presented the ProTractor, a simple and inexpensive ground-based imaging platform that demonstrates how large datasets of field images can be easily acquired. However, easy image acquisition with Go-Pro cameras mounted on a tractor leads to more difficulty when trying to localize individual rows of plants. To overcome this, we have proposed a row cropping algorithm that parses the images into a useful dataset. We have shown a proof-of-concept example of plant detection performed on this dataset that illustrates how the ProTractor can potentially be used as an end-to-end system for image-based phenotyping. In this section, we discuss the effectiveness of the ProTractor system, its limitations, and how we plan to improve upon this work in the future.

The row cropping algorithm showed strong precision and recall in the row detection task. The few errors mostly occurred due to erroneously detected weeds at the start and end of ranges. However, the row labelling task—in which exact row labels are assigned to each detected row—yielded a lower accuracy of 0.80. This is because the labelling task is subject to error propagation; if one row is labelled off-by-one, then all successive rows in the image sequence for that range will also be labelled off-by-one. In order to improve row labelling, we plan to incorporate a spatially-aware data structure that tracks row location and order. This data structure will allow interpolation within a range—where rows are expected but not detected. It will also enable optimization of the sequence of detected rows over all spatially pos-

sible combinations of candidate rows for a given range. For a given row, optimization will be carried out by taking into account plant pixel count, distance from neighbours, and slope of the fitted line segment. Thus, weeds that appear like rows at the start or end of a range could be rejected. The resulting spatially-aware row cropping algorithm should improve row detection and labelling accuracy.

The present approach for identifying plant rows has been evaluated for early season images appropriate for plant counting and assumes that plant rows are separated by soil. As future work, our approach could be adapted to other crops or later season images (e.g. after canopy closure) with different detection/segmentation techniques, such as convolution networks for segmentation [1].

The primary focus of this work has been on the physical ProTractor and its row cropping algorithm, and we have only provided a preliminary proposal for plant detection. A more fulsome investigation of plant detection and counting would be possible by leveraging the datasets the ProTractor generates, and is planned as future work. Our Faster-RCNN implementation showed good initial results achieving a *mAP* of 0.82 in plant detection, and an R^2 of 0.89 in plant counting. However, it does not perform well when there are overlapping plants — the model detects more plants than what the raters have annotated. The plants in the cropped row images are centred which is useful when annotating plants for the raters, because they can focus on one area. Another advantage of having the plants centred in the images is for measuring the vigor of the plants. The width and height of each row of plants are commonly used to score vigor, and this information could be easily determined using the images from our row processing software.

6. Conclusion

In this work, we have shown that it is possible and practical to collect and process per-row images of early season crops with consumer cameras on an existing tractor. We’ve demonstrated that rows can be consistently detected and that the resulting cropped row images are useful for counting *Brassica* seedlings in the field. Following the design of our ProTractor prototype, and our open source software for processing images, we hope that this work will help to enable routine collection of field images by plant scientists and result in new publicly available field imaging datasets for the computer vision in plant phenotyping community.

7. Acknowledgements

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund. We also thank Stacy Shirley from Agriculture and Agrifood Canada for construction of the ProTractor prototype.

References

- [1] S. Aich, A. Josuttis, I. Ovsyannikov, K. Strueby, I. Ahmed, H. S. Duddu, C. Pozniak, S. Shirtliffe, and I. Stavness. Deep-Wheat: Estimating Phenotypic Traits from Crop Images with Deep Learning. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018-January:323–332, 2018.
- [2] S. Aich and I. Stavness. Leaf counting with deep convolutional and deconvolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2080–2089, 2017.
- [3] J. L. Araus and J. E. Cairns. Field high-throughput phenotyping: The new crop breeding frontier. *Trends in Plant Science*, 19(1):52–61, 2014.
- [4] D. Ashlock, J. Brown, and Corns. K-Models Clustering, a Generalization of K-Means Clustering. *Intelligent Engineering Systems through Artificial Neural Networks*, 20:485–492, 2010.
- [5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.
- [6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [7] P. Burger, S. Liu, M. Hemmerlé, F. Baret, and B. Andrieu. Estimation of Wheat Plant Density at Early Stages Using High Resolution Imagery. *Frontiers in Plant Science*, 8(May):1–10, 2017.
- [8] L. Busemeyer, D. Mentrup, K. Mller, E. Wunder, K. Alheit, V. Hahn, H. P. Maurer, J. C Reif, T. Wrschum, J. Miller, F. Rahe, and A. Ruckelshausen. Breedvision - a multi-sensor platform for non-destructive field-based phenotyping in plant breeding. *Sensors (Basel, Switzerland)*, 13:2830–47, 03 2013.
- [9] W. Crum, O. Camara, and D. L. G. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, 2006.
- [10] D. Deery, J. Jimenez-Berni, H. Jones, X. Sirault, and R. Furbank. Proximal remote sensing buggies and potential applications for field-based phenotyping. *Agronomy*, 4(3):349–379, 2014.
- [11] A. Dobrescu, M. Valerio Giuffrida, and S. A. Tsiftaris. Leveraging multiple datasets for deep leaf counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2072–2079, 2017.
- [12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.
- [13] F. Gnadinger and U. Schmidhalter. Digital counts of maize plants by unmanned aerial vehicles (uavs). *Remote Sensing*, 9(6), 2017.
- [14] J. M. Guerrero, M. Guijarro, M. Montalvo, J. Romeo, L. Emmi, A. Ribeiro, and G. Pajares. Automatic expert system based on images for accuracy crop row detection in maize fields. *Expert Syst. Appl.*, 40(2):656–664, Feb. 2013.
- [15] M. Guijarro, G. Pajares, I. Riomoros, P. Herrera, X. Burgos-Artizzu, and A. Ribeiro. Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1):75 – 83, 2011.
- [16] K. N. Harker, J. ODonovan, E. G. Smith, E. Johnson, G. Peng, C. J. Willenborg, R. Gulden, R. Mohr, K. S. Gill, J. Weber, et al. Canola growth, production, and quality are influenced by seed size and seeding rate. *Canadian Journal of Plant Science*, 97(3):438–448, 2016.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [18] P. V. C. Hough. Method and means for recognizing complex patterns. united states patent 3069654, December 1962.
- [19] R. Ji and L. Qi. Crop-row detection algorithm based on Random Hough Transformation. *Mathematical and Computer Modelling*, 54(3-4):1016–1020, 2011.
- [20] J. A. Jimenez-Berni, D. M. Deery, P. Rozas-Larraondo, A. T. G. Condon, G. J. Rebetzke, R. A. James, W. D. Bovill, R. T. Furbank, and X. R. Sirault. High throughput determination of plant height, ground cover, and above-ground biomass in wheat with lidar. *Frontiers in plant science*, 9:237, 2018.
- [21] X. Jin, S. Liu, F. Baret, M. Hemerlé, and A. Comar. Estimates of plant density of wheat crops at emergence from very low altitude UAV imagery. *Remote Sensing of Environment*, 198:105–114, 2017.
- [22] K. Krippendorff. *Content analysis: An introduction to its methodology*, 3rd edition. Thousand Oaks: SAGE Publications, 2013.
- [23] S. U. Lee, S. Y. Chung, and R. H. Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2):171 – 190, 1990.
- [24] V. Leemans and M. F. Destain. Line cluster detection using a variant of the Hough transform for culture row localisation. *Image and Vision Computing*, 24(5):541–550, 2006.
- [25] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.
- [26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [27] T. Liu, W. Chen, W. Guo, X. Zhu, C. Sun, and W. Wu. Automated image-processing for counting seedlings in a wheat field. *Precision Agriculture*, 17(4):392–406, 2015.
- [28] M. Minervini, A. Fischbach, H. Scharr, and S. Tsiftaris. Plant phenotyping datasets, 2015.
- [29] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsiftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition letters*, 81:80–89, 2016.
- [30] M. Montalvo, G. Pajares, J. M. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. J. Ruz, and J. M. Cruz. Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Syst. Appl.*, 39(15):11889–11897, Nov. 2012.
- [31] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP*

- International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.
- [32] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.
- [33] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6656–6664, 2017.
- [34] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.
- [35] J. Ribera, Y. Chen, C. Boomsma, and E. J. Delp. Counting plants using deep learning. *2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017 - Proceedings*, 2018-Janua:1344–1348, 2018.
- [36] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016.
- [37] A. Shafiekhani, S. Kadam, F. B. Fritschi, and G. N. DeSouza. Vinobot and vinoculer: Two robotic platforms for high-throughput field phenotyping. In *Sensors*, 2017.
- [38] D. Shrestha, B. Steward, and S. Birrell. Video processing for early stage maize plant detection. *Biosystems Engineering*, 89(2):119 – 129, 2004.
- [39] H. T. Sjøgaard and H. J. Olsen. Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 38(2):141–158, 2003.
- [40] A. Stein, C. J. ter Braak, G. W. van der Heijden, S. A. Hiremath, and F. K. van Evert. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture*, 100:41–50, 2013.
- [41] J. Svensgaard, T. Roitsch, and S. Christensen. Development of a Mobile Multispectral Imaging Platform for Precise Field Phenotyping. *Agronomy*, 4(3):322–336, 2014.
- [42] L. Tzuta. Labelimg, 2015.
- [43] I. Vidović, R. Cupec, and Ž. Hocenski. Crop row detection by global energy minimization. *Pattern Recognition*, 55:68–86, 2016.
- [44] N. Virlet, K. Sabermanesh, P. Sadeghi-Tehran, and M. Hawkesford. Field scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Functional Plant Biology*, 44:143–153, 11 2016.
- [45] N. Virlet, K. Sabermanesh, P. Sadeghi-Tehran, and M. J. Hawkesford. Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Functional Plant Biology*, 44(1):143–153, 2017.
- [46] U. Weiss and P. Biber. Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and Autonomous Systems*, 59(5):265–273, 2011.
- [47] J. W. White, P. Andrade-Sanchez, M. A. Gore, K. F. Bronson, T. A. Coffelt, M. M. Conley, K. A. Feldmann, A. N. French, J. T. Heun, D. J. Hunsaker, et al. Field-based phenomics for plant genetics research. *Field Crops Research*, 133:101–112, 2012.
- [48] J. T. Yagushi, D. S. Costa, and J. d. B. França-Neto. Saturated salt accelerated aging and computerized analysis of seedling images to evaluate soybean seed performance. *Journal of Seed Science*, 36:213 – 221, 06 2014.
- [49] G. Yantai, K. N. Harker, H. R. Kutcher, R. H. Gulden, B. Irvine, W. E. May, and J. T. O’Donovan. Canola seed yield and phenological responses to plant density. *Canadian journal of plant science*, 96(1):151–159, 2016.
- [50] Z. Yu, Z. Cao, X. Wu, X. Bai, Y. Qin, W. Zhuo, Y. Xiao, X. Zhang, and H. Xue. Automatic image-based detection technology for two critical growth stages of maize: Emergence and three-leaf stage. *Agricultural and Forest Meteorology*, 174-175:65 – 84, 2013.
- [51] H. Zhang, B. Chen, and L. Zhang. Detection Algorithm for Crop Multi-Centerlines Based on Machine Vision. *Transactions of the ASABE*, 51(3):1089–1097, 2008.