remote sensing

MDPI

Article

An Automated Framework for Plant Detection Based on Deep Simulated Learning from Drone Imagery

Benyamin Hosseiny [1], Heidar Rastiveis [1] and Saeid Homayouni [2,*]

[1] Department of Photogrammetry and Remote Sensing, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1417466191, Iran; ben.hosseiny@ut.ac.ir (B.H.); hrasti@ut.ac.ir (H.R.)

[2] Centre Eau Terre Environnement, Institut National de la Recherche Scientifique (INRS), Quebec, QC G1K 9A9, Canada

* Correspondence: saeid.homayouni@ete.inrs.ca

Received: 25 September 2020; Accepted: 22 October 2020; Published: 27 October 2020

Abstract: Traditional mapping and monitoring of agricultural fields are expensive, laborious, and may contain human errors. Technological advances in platforms and sensors, followed by artificial intelligence (AI) and deep learning (DL) breakthroughs in intelligent data processing, led to improving the remote sensing applications for precision agriculture (PA). Therefore, technological advances in platforms and sensors and intelligent data processing methods, such as machine learning and DL, and geospatial and remote sensing technologies, have improved the quality of agricultural land monitoring for PA needs. However, providing ground truth data for model training is a time-consuming and tedious task and may contain multiple human errors. This paper proposes an automated and fully unsupervised framework based on image processing and DL methods for plant detection in agricultural lands from very high-resolution drone remote sensing imagery. The proposed framework's main idea is to automatically generate an unlimited amount of simulated training data from the input image. This capability is advantageous for DL methods and can solve their biggest drawback, i.e., requiring a considerable amount of training data. This framework's core is based on the faster regional convolutional neural network (R-CNN) with the backbone of ResNet-101 for object detection. The proposed framework's efficiency was evaluated by two different image sets from two cornfields, acquired by an RGB camera mounted on a drone. The results show that the proposed method leads to an average counting accuracy of 90.9%. Furthermore, based on the average Hausdorff distance (AHD), an average object detection localization error of 11 pixels was obtained. Additionally, by evaluating the object detection metrics, the resulting mean precision, recall, and F1 for plant detection were 0.868, 0.849, and 0.855, respectively, which seem to be promising for an unsupervised plant detection method.

Keywords: plant detection; deep learning; faster R-CNN; ResNet-101; drone imagery; precision agriculture

1. Introduction

Smart and precision agriculture (PA) for managing the arable lands are presently essential agronomy practices to improve food productivity and security and environment protection in a sustainable agriculture context [1–3]. These advancements help agronomists and field experts to use modern technologies instead of traditional field monitoring, which are laborious and time-consuming processes [4]. Therefore, developing robotic applications and data processing methods play a crucial role in modern agronomy. Generally, PA involves optimizing farm inputs such as fertilizers, seed, and fuel to improve crop production. Consequently, many technological advances

in data collection, processing, analysis, and sensor design are involved in PA [5,6]. These technologies help farmers and farm managers go toward smart agriculture, using new technologies to increase the quality of the products and enhance people's lifestyle [2].

In recent years, digital sensors, such as high-resolution red/green/blue spectral (RGB) band cameras, multi- and hyperspectral imagery, and thermal images, have been efficiently used for data collection in PA [5]. These sensors are mounted on different platforms, such as satellites, aircraft, drones (or unmanned aerial vehicles; UAVs), or terrestrial platforms.

Among these platforms, drones can be one of the most suitable platforms for agricultural or farming tasks because of their ability to provide very high-resolution images, flexibility, easy-to-access sensors, and relatively low-cost data collection [7,8]. They are cost-effective and easy-to-carry platforms and usually can fly in cloudy weather because of low altitude. They can also provide very high temporal and spatial resolution data to monitor farmlands on a per-plant basis [9]. Compared to the ground-based platforms, drones can cover large areas in a comparably short amount of time, while they do not affect the fields through soil compaction or damaging vegetation as ground vehicles do. These capabilities have made drone imagery systems powerful tools in various agro-environmental remote sensing applications [9–13]. In this regard, various objects and complex textures, noise, lighting conditions, and geometrical distortion of image scenes are some challenges in the automatic processing of remote sensing data.

Thanks to the outstanding performance of deep learning (DL) methods, in general, and the convolutional neural networks (CNNs), in particular, to solve complex problems in the field of computer vision and pattern recognition, they have been widely used in various applications [4]. Applications of DL in remote sensing have been emerging since 2014 [14]. Ever since then, many state-of-the-art DL-based algorithms have been developed to solve remote sensing problems such as change detection [15], classification [16], and target and anomaly detection [17], which are challenging problems in PA. Comprehensive reviews on using DL techniques for agricultural applications have been provided by [18] and [19].

Determining plant numbers and exact locations in an agricultural field can provide rich statistical information about crop products, leading to a better estimation of plant density and possibly direct yield estimation. These results are essential for a more accurate assessment of the product's health and final yield maturity. Among several studies, only a few research works have focused on the precise counting of vegetation or fruits. However, there has been rapid progress in applying DL to the task of object detection in PA. For example, Dijkstra et al. [20] proposed a fully CNN based on U-Net, namely CentroidNet, for crop localization and counting from UAV images. Then it generated the segmented images as the output, which contained the centers of the plants. Wu et al. [21] proposed a system to detect and estimate the number of rice seedlings from RGB UAV imagery by proposing two fully CNNs. The first network was responsible for segmenting rice seedling areas and non-rice seedling areas in each image, while the second one was designed and trained to estimate the distribution of rice seedlings in each image and generate a density map. Ribera et al. [22] developed a new loss function based on Hausdorff distance [23] to estimate the objects' location based on their central points. They implemented it in a fully CNN and evaluated it on drone RGB images for detecting plants. Bellocchio et al. [24] proposed a weakly supervised method to count fruits. They developed a novel CNN architecture based on ResNet-101 and proposed two loss functions, the presence-absence classifier (PAC) and spatial consistency. The network processes the input image at three scales. Recently, Osco et al. [25] proposed a CNN architecture to count and localize citrus trees with high density from high-resolution multispectral UAV images. Their proposed network was based on fully convolutional network (FCN) models, and the output was a density map that estimates the location of objects.

Over the past decade, many studies have been conducted to improve the object detection problem using DL methods. These efforts resulted in several object detection frameworks, including different versions of region-based CNNs (R-CNNs; fast R-CNN, faster R-CNN, and mask R-CNN) [26–29] and you only look once (YOLO; v1-v3) [30–32]. The central part of these frameworks consists of robust object detectors that are mostly based on CNNs with complex architectures, such as the Oxford Visual

Geometry Group Network (VGGNet) [33] and Residual Network (ResNet) [34]. Similarly, precision counting of objects gained much attention by significant accuracy improvements of the object detection frameworks. In this regard, various object detection algorithms have been developed and used in PA. For example, Sa et al. [35] developed a DL fruit detection system, based on faster R-CNN, from a multispectral sensor and Microsoft Kinect 2 data. They used the VGG-16 model, and the pre-trained ImageNet dataset initialized the trainable parameters. In addtion, Zhou et al. [36] developed and implemented a panicle detection and counting system for UAV RGB imagery of rice fields based on an improved region-based fully CNN.

Object detection architectures usually include millions of trainable parameters, which can be the main drawback of the DL-based models. One prevalent solution to this issue is using pre-trained models and then fine-tuning them through transfer learning by training the parameters based on the primary dataset [19]. However, providing diverse and appropriate training data with ground truth is still a big challenge and a time-consuming task [35,37,38]. Pre-trained models can be trained by publicly available image datasets, such as PASCAL Visual Object Classes (PASCAL VOC), Microsoft Common Objects in COntext (COCO) [39], and ImageNet [40], which include thousands of common object classes and millions of images. These models and datasets are available to developers for model training or for benchmarking object recognition algorithms. Another standard solution is data augmentation by manipulating the training data to increase their diversity and quantity. Data augmentation techniques mostly include rotation, scale, changing brightness, inverse mirroring [36,41], and synthetic image generation. Rahnemoonfar and Sheppard [37] generated about 24,000 simulated images consisting of tomato shapes and textures in different sizes for training a deep object detector for counting and localizing tomatoes.

Despite all the research and development efforts and existing literature in this field, there is still the need for a framework that can efficiently and automatically detect and extract plants. This paper proposes an automated and fully unsupervised framework based on image processing techniques and DL methods for plant detection in agricultural lands from very high-resolution drone remote sensing imagery. The main objective of this paper is to propose an automatic framework to generate an unlimited amount of simulated training data from the input image. To this end, to generate many plant objects, various single plants are rotated and scaled and then are implanted with variable brightness in the simulated bare soil patches. This approach is completely automatic without any user intervention, thus eliminating the need to gather ground truth for training data as a time-consuming and laborious job. These data are then used as training data in the faster R-CNN framework, a robust and reliable object detection method with the ResNet-101 convolutional network, to detect and count the plants in high-resolution drone RGB images.

2. Proposed Framework

Figure 1 shows the workflow of the proposed method for plant detection from high-resolution drone imagery. As can be seen from this figure, the proposed algorithm works in three consecutive steps: (i) automated training data generation, (ii) modeling based on a DL object detection framework, and (iii) plant detection.

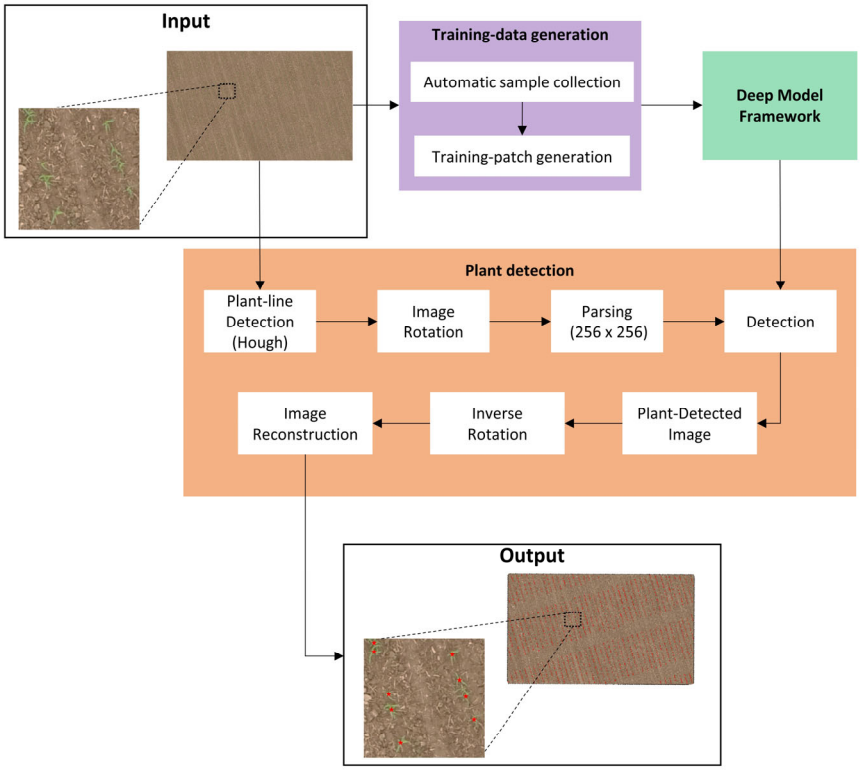


Figure 1. The workflow of the proposed method for automated plant detection.

2.1. Training Data Generation

Training data with high quantity, quality, and diversity are essential to obtain acceptable results by using DL models. However, manually generating an appropriate amount of training data can be time-consuming. In our proposed method, an automated method for generating training data for plant detection is used. It is worth noting that the plant sizes are too small compared to the main image's size. Therefore, the image is divided into several small non-overlapping patches with smaller dimensions than the main image (i.e., 256 × 256 pixels).

2.1.1. Automatic Sample Collection

For generating simulated training data, candidate objects or plants are firstly separated from the background area. Many studies have exploited various vegetation indices for automatically separating vegetation from the background bare soil in fields [5,7,42]. These indices are usually computed from visible RGB or multispectral bands, such as near-infrared (NIR). The excess green index (ExG) based on the RGB data in the visible spectrum is one of the most used vegetation indices (Equation (1)) [4,7,9,38]:

$$ExG = 2G - R - B \quad (1)$$

where *R*, *G*, and *B* are the image channels corresponding to red, green, and blue spectral bands. Once the ExG image is calculated, a binary C-means clustering [43] is used to discriminate the plants from the background, i.e., soil, stones, etc. Although thresholding techniques, such as Otsu's method, can be used to separate vegetative areas from the background in the ExG image, in this research, the C-means algorithm is applied because of its robust results and short processing time. However, after running the algorithm multiple times, the most frequent result is selected as the optimum output. Figure 2 shows the background masking process's step-by-step results based on the ExG and binary clustering for a sample area.

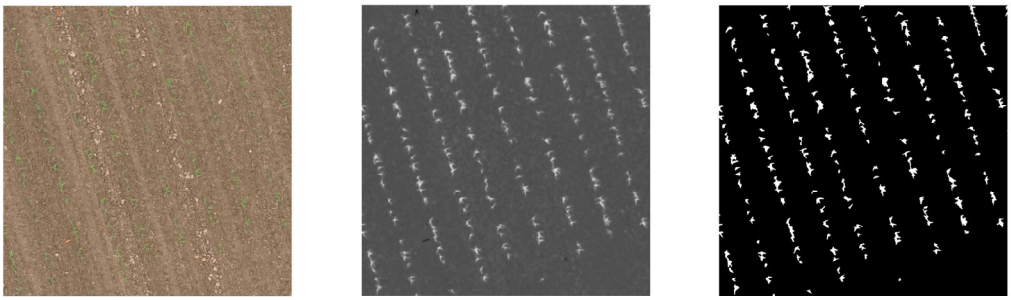


Figure 2. Separating plants from the background by the excess green (ExG) index and binary clustering: (**a**) Input image (**b**) ExG gray-level image, (**c**) binary clustered output of the ExG image.

The resulting binary image includes several objects of various sizes and shapes. Among them, plants are the majority of the objects. However, several false objects or overlapping areas, including two or more plants, may appear. As the single plants' areas are almost similar, analyzing the objects' frequency histogram may be useful for extracting several sample plants. Figure 3 shows the frequency histogram of the extracted objects in the sample image shown in Figure 2, which includes a wide range of area sizes.

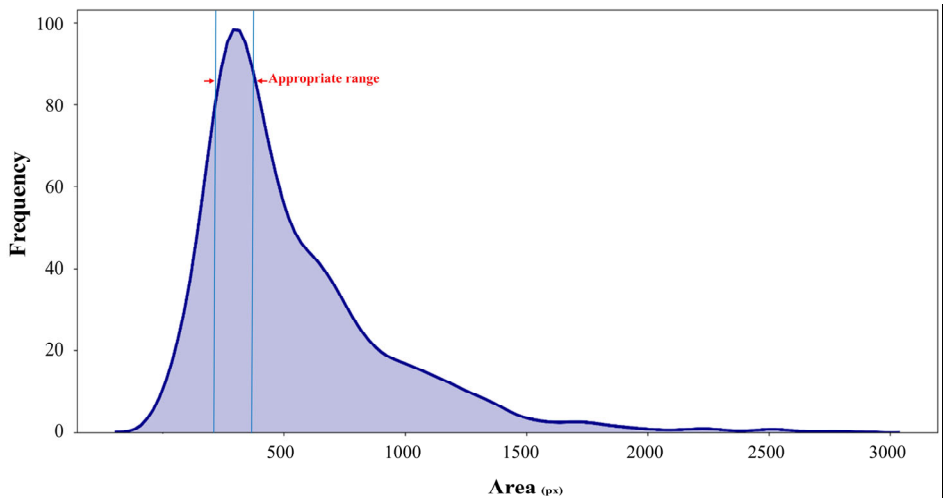
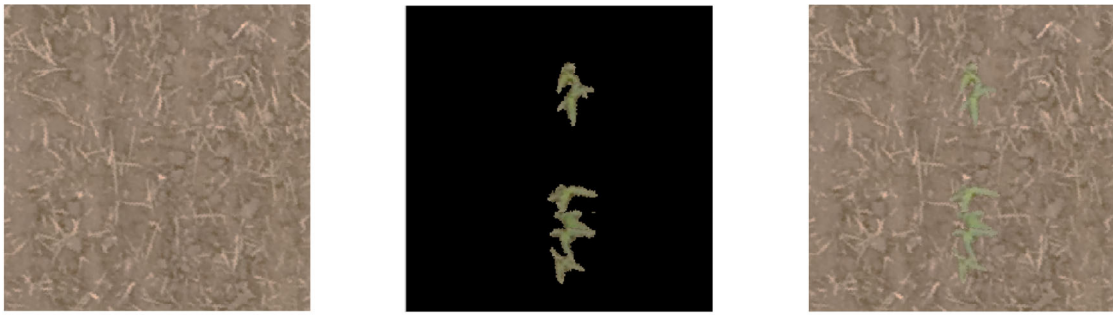


Figure 3. Frequency histogram of the areas of the extracted objects from the binary clustered image. Vertical lines determine the minimum and maximum areas for selecting single plants.

Tiny objects with small areas are suspected to be false alarms, while the larger objects could be the overlapping plants that appeared as a single big object. Therefore, the most frequent areas, which appeared as the histogram apex, have a higher confidence level for being the area of single plants. We adapted the interquartile range (IQR) method for separating these three classes of object sizes. IQR is a popular method for detecting outliers or abnormal signals. An appropriate range for extracting single plants is located between $M–k \times IQR$ and $M–k \times IQR$ where M is the most frequent area based on the histogram, IQR is the interquartile range, and k is constant, which can be defined as a scale factor of radial distance from M with respect to IQR in the histogram. An empirical approach is usually used to determine the value of k, which depends on the signals and the application. However, about 50% of the samples are selected when k is equal to 0.5. As a result, objects located in this range of areas are determined as normal-sized objects and are extracted as candidate single plant samples used in training patch generation.

2.1.2. Training Patch Generation

This module's main objective is to generate an unlimited amount of training images (patches), including several known plants, to train the proposed DL-based object detection model. They are created in three main steps. First, the image background, which is bare soil, is extracted from the binary clustered image. Second, some of the extracted single plants from the previous section (Section 2.1.1) are randomly selected. These single plants are implanted in the generated background image in the third step. It is worth mentioning that plant overlap should be considered to improve the deep object detector model's ability to extract overlapping plants, which may be seen in real-world cases. Other relevant data augmentation methods such as scale, rotation, and brightness are considered to increase the patches (see Section 3.2 for more details). Note that these training patches' sizes are the same as the input patches, 256 × 256 pixels. Figure 4 shows an example of implementing these three steps for a sample training patch.



(a) (b) (c)

Figure 4. Steps of training image patch generation: (**a**) A sample generated background image; (**b**) extracted and implanted single plants with normal shapes; (**c**) final generated training patch with a combination of the generated background and single plant images.

2.2. Deep Model Framework

A robust object detection system is required to extract plants even if they are highly overlapped or have various scales. The CNNs are basically a group of trainable kernels that are convolved with an n-dimensional input signal. The main characteristic of a deep CNN is to explore and find nonlinear features, which are embedded in the input signal and cannot be extracted by simple linear transformations. The whole network and its parameters are trained by the backpropagation process [44].

In this paper, the faster R-CNN [28], as a robust and fast object detection framework, is employed with the ResNet [34] architecture as its deep backbone model for extracting plants. Faster R-CNN consists of two main modules: the region proposal network (RPN) and classification. The main sections of the faster R-CNN framework are shown in Figure 5. As can be seen from this figure, the input image's feature map is firstly generated based on a deep CNN, and then it is fed to the RPN to detect objects' presence probability with the softmax layer and their bounding box with regression. After the RPN, the proposed regions are presented with different sizes. Therefore, a region of interest (ROI) pooling section is provided to transform the feature maps into a unique shape. Finally, the RPN outputs are imported into a classifier, with a fully connected (FC) layer, to classify and label every detected object [19,28].

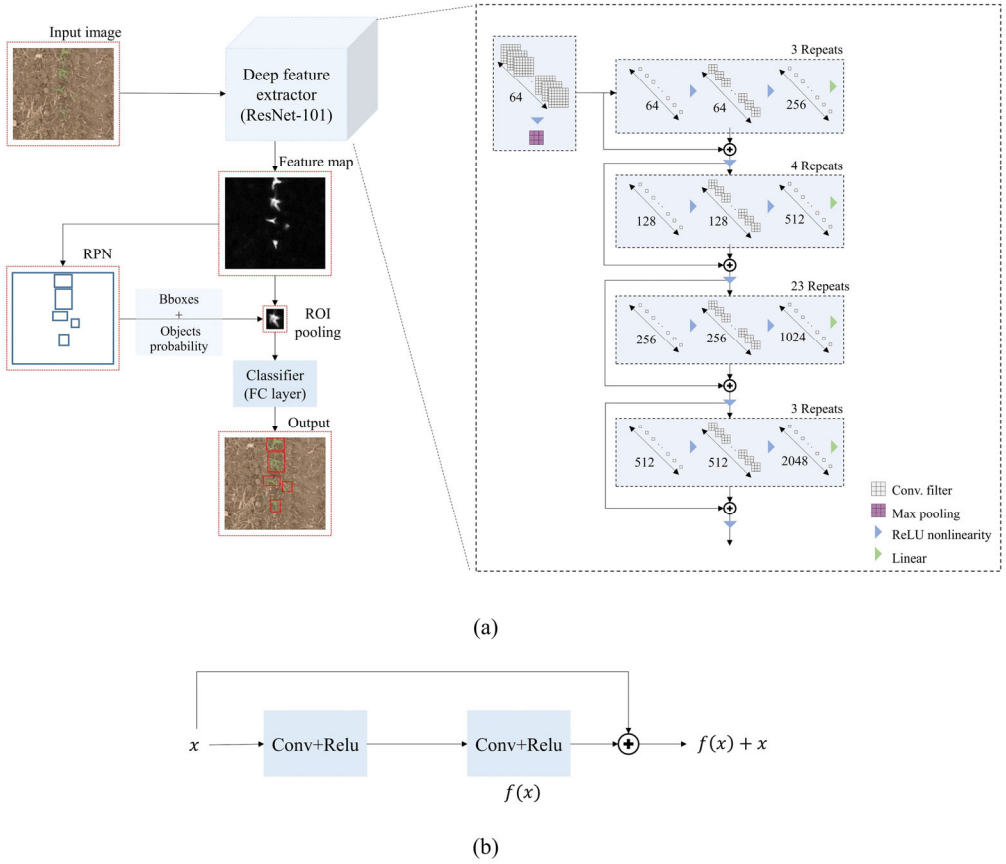


Figure 5. (**a**) Faster regional convolutional neural network (R-CNN) object detection framework that uses ResNet-101 as a deep feature extractor; (**b**) the structure of a residual sample block in the ResNet listed in Table 1.

In this paper, the ResNet-101 convolutional network is used as the deep feature extractor backbone of the faster R-CNN. The summary of the architecture of ResNet-101 is provided in Table 1. The ResNet architecture's first layer consists of a common convolutional layer and max-pooling operator with the stride of two, while the other layers are the residual blocks. In general, a ResNet architecture is made of five convolutional blocks: one simple convolutional block at the beginning and four subsequent residual blocks. Figure 5b shows a residual block's components composed of two convolutional layers with the rectified linear unit (ReLU) activation function. A residual block's output is acquired by adding the input feature array to the second output layer's residual block. The combination of faster R-CNN and ResNet architecture results in one of the most popular and accurate object detection frameworks [2]. To accelerate the model training, we used the model's pre-trained weights based on the COCO dataset [45] and fine-tuned our model by training the generated dataset through the transfer learning.

Table 1. ResNet-101 deep feature extractor layers' characteristics.

Convolutional Block	Number of Repeats	Kernels	Number of Filters
#1 Simple convolutional block	1	(7 × 7)	64
		(3 × 3)	Max pooling
#2 Residual block	3	(1 × 1)	64
		(3 × 3)	64
		(1 × 1)	256
#3 Residual block	4	(1 × 1)	128
		(3 × 3)	128
		(1 × 1)	512

		(1 × 1)	256
#4 Residual block	23	(3 × 3)	256
		(1 × 1)	1024
		(1 × 1)	512
#5 Residual block	3	(3 × 3)	512
		(1 × 1)	2048

2.3. Plant Detection

As described in the previous sections, a deep object detection framework is trained for detecting plants from the input image using generated training patches. Therefore, the dimension of the input image should be the same as the training patch dimension. In other words, the main image should be divided into several tiles with the dimension equal to the training patch dimension. In this case, inclined crop rows may result in several split plants during the tiling process. Therefore, in this algorithm, the plant cultivation direction is detected at first. The image is then rotated so that the crop rows are aligned either vertically or horizontally. For this purpose, the Hough transform (HT) [46] is applied to the binary clustered images to extract the crop rows. The HT is one of the most widely used methods for line detection on images [9,38,47]. It generates a 2-D histogram with the size of an $M \times N$ array in the Hough space, where M is the different values of the radius ρ and N is the different values of the angle θ. Each of the non-zero pixels in (ρ, θ) space can be considered as potential line candidates in the image space. The local maxima values in the histogram indicate the parameters of the most probable lines [46]. The direction of the longest line is estimated as the main direction of the crop row. After rotating the image based on the detected direction, it is parsed into small patches. Each image patch is fed into the trained object detector, and the predicted map is inversely rotated to generate the final map with the size of the input image.

2.4. Evaluation Metrics

To evaluate the proposed DL framework quantitatively, we used three main metrics, including mean absolute error (MAE) and accuracy (ACC) [21,22,45] for evaluating the plant counting errors, and the average Hausdorff distance (AHD) [22,23] for estimating the localization error of the results, which can be interpreted as the average location error. These metrics can be calculated using the following equations:

$$MAE = \frac{1}{N}\sum_{i}^{N}|y_i - \hat{y}_i| \tag{2}$$

$$ACC = 1 - \frac{1}{N}\sum_{i}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \tag{3}$$

$$AHD = \frac{1}{|\hat{P}|}\sum_{\hat{p}\in\hat{P}}\min_{p\in P} d(\hat{p},p) + \frac{1}{|P|}\sum_{p\in P}\min_{\hat{p}\in\hat{P}} d(p,\hat{p}) \tag{4}$$

where y_i and $\hat{y}_i$, respectively, represent the ground truth count and predicted count for the i^{th} image patch, N is the number of total image patches, and P and $\hat{P}$ are the sets of points representing the ground truth and predicted location of the center of the objects, respectively. $|P|$ and $|\hat{P}|$ are the number of points in P and $\hat{P}$ as well, and $d(\cdot,\cdot)$ is a distance function that, in this study, was determined as Euclidean distance.

3. Experiments and Results

3.1. Drone Imagery

This research used two RGB drone images collected at a flying altitude of 30 m over two cornfields near Shooshtar City, Iran. In both of the fields, the corn seeds were cultivated in regular crop rows. The size of each image by the digital camera was around 20 megapixels. The camera was

mounted on a DJI Phantom 4 Pro system (www.dji.com). All of the main imaging parameters are gathered in Table 2. To simplify the dataset, we cropped a 4000 × 6000 area from the first image and 4000 × 5000 from the second image as the study areas. The full views of the selected sample areas from these two datasets are shown in Figure 6. As can be seen from this figure, planting, soil, and illumination conditions of these datasets are different. As a result, the robustness of the proposed methodology could be evaluated.

Table 2. Image acquisition details of the used datasets (GSD: Ground sampling distance; ISO: International Organization of Standardization).

Camera Model	Flight Height (m)	Spectral Bands	GSD (cm)	ISO	Exposure Time (s)	Focal Length (mm)
FC6310	30	R-G-B	0.8	200	1/1250	9

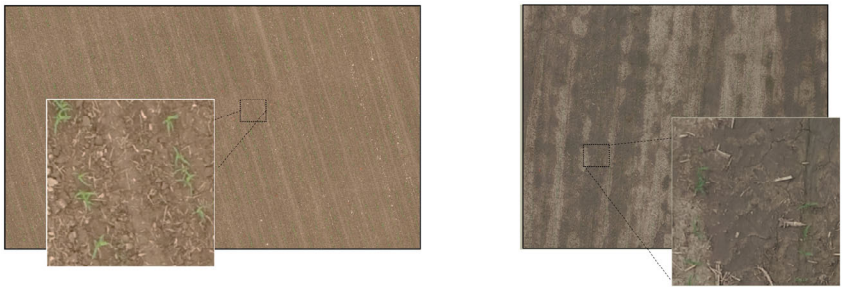


Figure 6. Full view of the investigated datasets: (**a**) First dataset and (**b**) second dataset.

Figure 7 depicts the phenological stages of a cornfield during a growing season. Based on these stages, in our first investigated image, most of the corn plants were in the second leaf collar stage (V2), while some plants were in the first (V1) and third (V3) stages. In the second image, the plants' phenological stages were a combination of emergence (VE), V1, and V2 stages; however, most of the plants belonged to the first stage (V1). In general, the drone imagery was taken in the earlier stage of crop emergence. In this way, the leaf overlaps were minimal and could lead to a more accurate single plant detection.

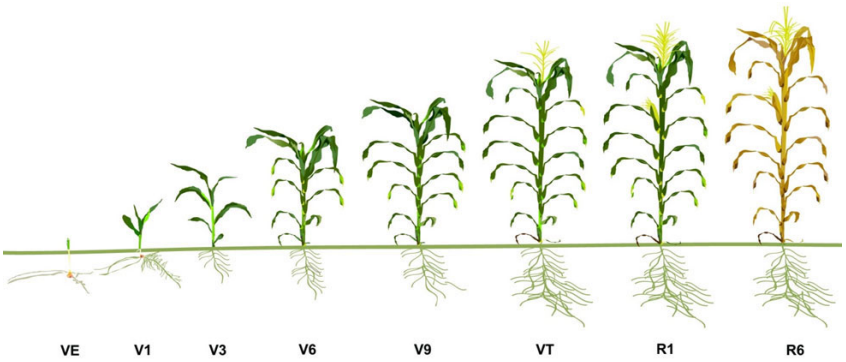


Figure 7. Phenological stages of cornfield growth (Figure adapted from Figure 1 in [48]).

3.2. Implementation

After extracting the ExG descriptor and performing the C-means algorithm in the training patch generation step, the candidate sample single plants were extracted by adapting the IQR method, considering a constant value of $k = 0.2$. It is worth mentioning that increasing the value of k would result in more variable objects with diverse areas. Conversely, by choosing a smaller value, more similar objects would be obtained. Then, 800 training patches with dimensions of 256 × 256 pixels were generated using these samples separately in each dataset. Among these samples, 200 samples

were generated based on the normal-sized extracted objects, 400 samples were generated by simulating targets with different scales, e.g., 50% bigger than normal-shaped targets, 200 patches were generated with different brightness values, and the 200 remaining patches were simulated with rotated target orderings in the background. Table 3 summarizes the simulated training patch descriptions.

Table 3. Simulated training sample description.

Simulation Type	Generated Samples
Normal-sized objects	400
Scaled objects (×0.5)	50
Scaled objects (×0.75)	50
Scaled objects (×1.5)	50
Scaled objects (×1.75)	50
Rotated object ordering	100
Object brightness changed (brightness increased between 10 and 30)	50
Object brightness changed (brightness increased between 30 and 10)	50
Total:	**800 samples**

Figure 8 illustrates a number of the generated training patches for each dataset. As can be seen, changing scale, rotation, or brightness helped create diverse training patches. These patches were then used to train the DL network.

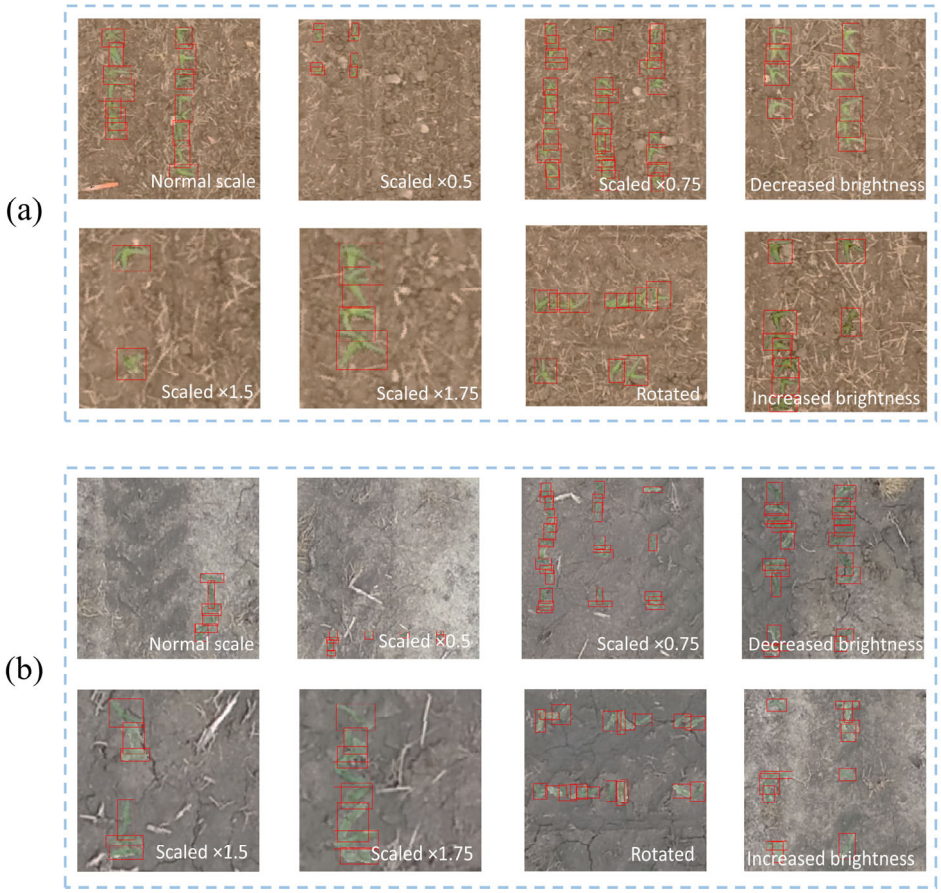


Figure 8. Some instances of the generated images, based on the proposed method for the first (**a**) and the second (**b**) datasets.

After training the network, to extract all plants from the datasets, the image scene was first rotated to align the crop rows vertically. In this case, crop rows' orientation was estimated to be −17.1 and +4.8 degrees in the first and second image sets, respectively. Figure 9 summarizes the alignment processing of the first dataset. Finally, the image scenes were divided into 256 × 256 small patches. As a result, there were 442 and 328 patches for the first and the second datasets.

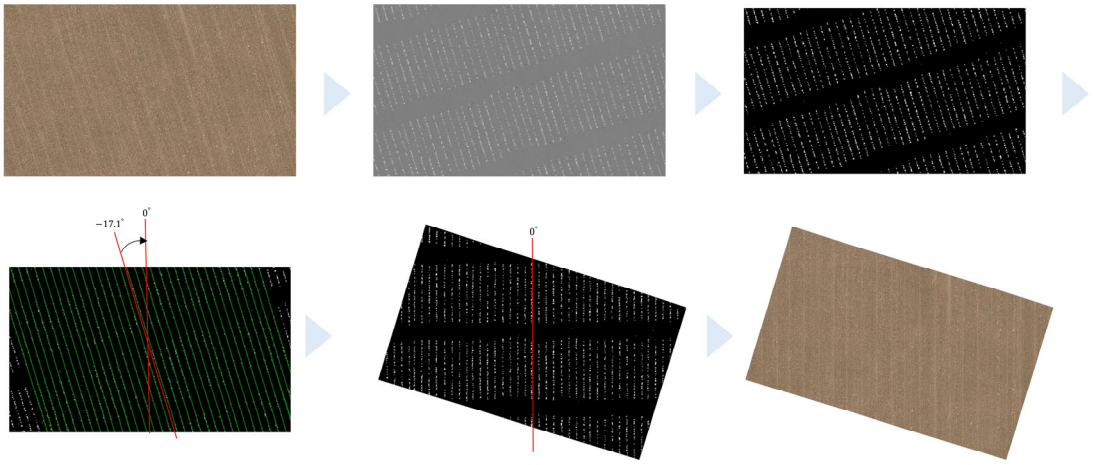


Figure 9. The image alignment process, based on the crop row direction.

3.3. Results

For each dataset, the faster R-CNN object detector was trained with 800 simulated samples (Figure 8) for 50 epochs. Finally, each parsed image patch was fed to the trained model to extract plants and their counts. Figures 10 and 11 show the final visual results, which are the detected plants' locations in the first and second drone images, respectively.

Figure 10. (**a**) Detected plants in the first image: (**b**) Instances of the extracted image patches. Red stars are the centers of predicted plants according to the object detector.

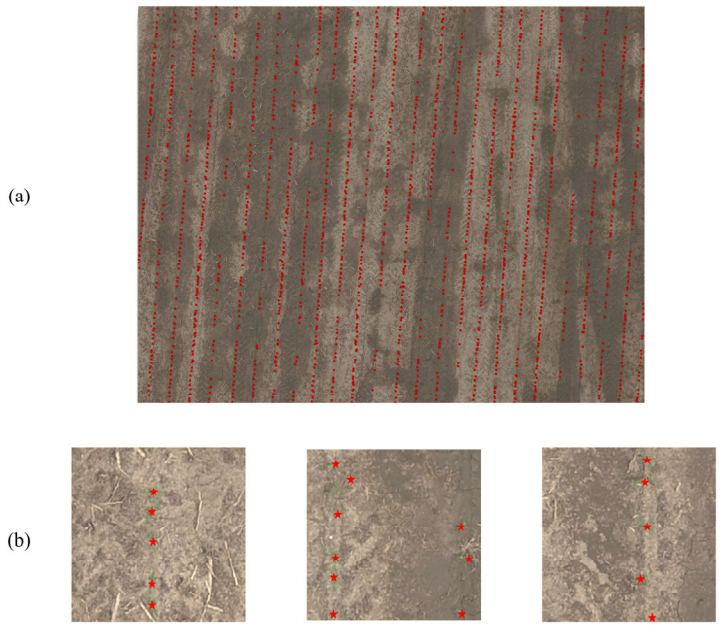


Figure 11. (**a**) Detected plants in the second image: (**b**) Instances of the extracted image patches. Red stars are the centers of predicted plants according to the object detector.

3.4. Accuracy Assessment

In this paper, the validity of the results was evaluated by comparing the extracted plants with manually extracted plants. Figure 12 visually compares the manually extracted plants with the estimated plant location by the proposed algorithm for many patches from both datasets. In these figures, blue stars are the objects' center ground truth points, and red stars are the trained model's predicted centers. As can be seen, most of the estimated points are located close to the ground truth points. Moreover, most of the plants in these patches were successfully extracted.

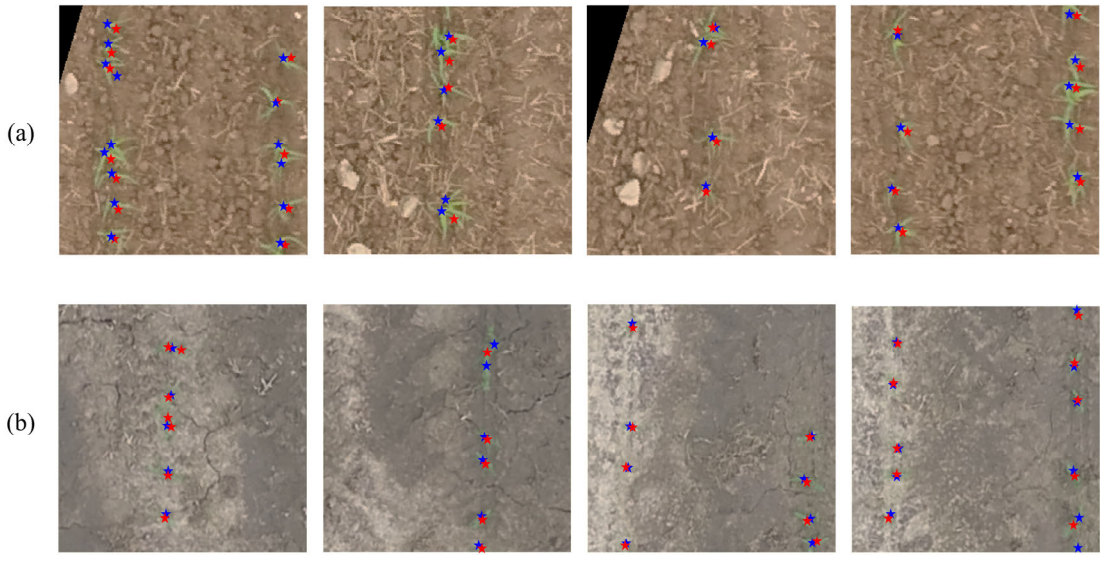


Figure 12. Visual comparison of detected plants with their ground truth locations for the extracted patches from (**a**) first image (**b**) second image. Blue stars are ground truth points, and the red ones are predicted based on the model.

The counting and localization errors were calculated based on the MAE, ACC, and AHD metrics, provided in Table 4. As can be seen, the model detects and counts 92.4% and 89.4% of the objects,

with an MAE of 0.253 and 0.571, for each patch and the first and the second datasets, respectively. The objects' localization errors based on the AHD metric were 11.51 in the first dataset and 10.98 pixels for the second one.

Table 4. Plant counting and detection evaluation results for investigated image scenes (MAE: Mean absolute error; AHD: Average Hausdorff distance).

Dataset	Number of Patches	Average Number of Plants per Patch	MAE	Accuracy (%)	Mean AHD (pixels)
Dataset 1	442	6.65	0.253	92.4	11.51
Dataset 2	328	5.68	0.571	89.4	10.98

Figure 13 represents the difference between the ground truth count and each image patch's predicted count for the investigated scenes. It can be observed that in both datasets, most of the points are zero, which shows that the trained model by simulated patches was able to extract and count plants accurately. However, in several patches, a few underestimated cases were observed in highly overlapping areas.

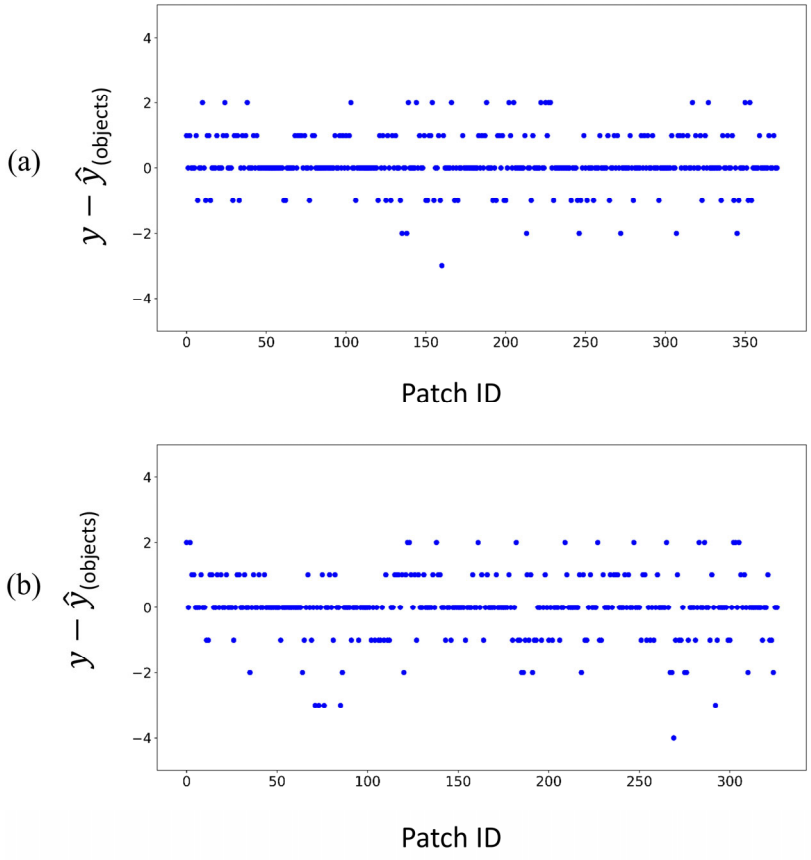


Figure 13. Difference between ground truth and predicted counts for each image patch (**a**) for first and (**b**) second datasets.

Figure 14 shows the histogram of normalized counting errors of all the patches from both datasets. As is evident, more than 400 of the patches had a normalized error of less than 0.1 (or 10%). For a more in-depth investigation of the proposed framework's drawbacks, we evaluated the patches with an error greater than 0.50. Fortunately, the total number of these patches was 17 (2.4% of total), and they are shown in Figure 15. As can be seen, most of the proposed framework faults happened in the cases containing very small or incomplete plants, especially on the edges of the main image scene. Additionally, most of the patches with the highest error (i.e., 1) contained one or two plants

that could not be detected by our object detector framework that may have been due to their very small size.

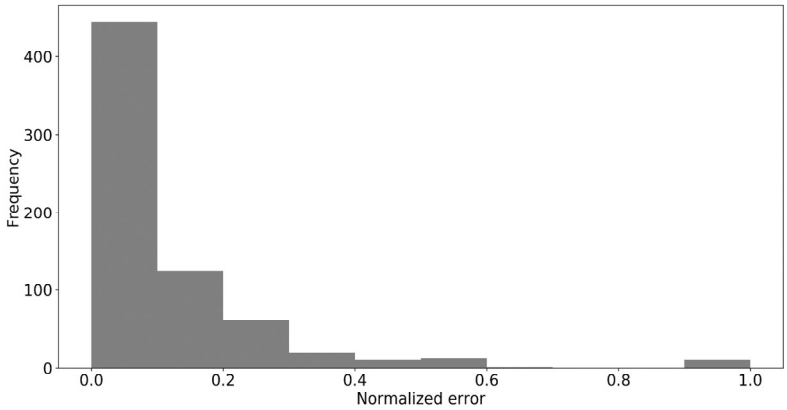


Figure 14. Histogram of normalized counting error for the image patches in both datasets.

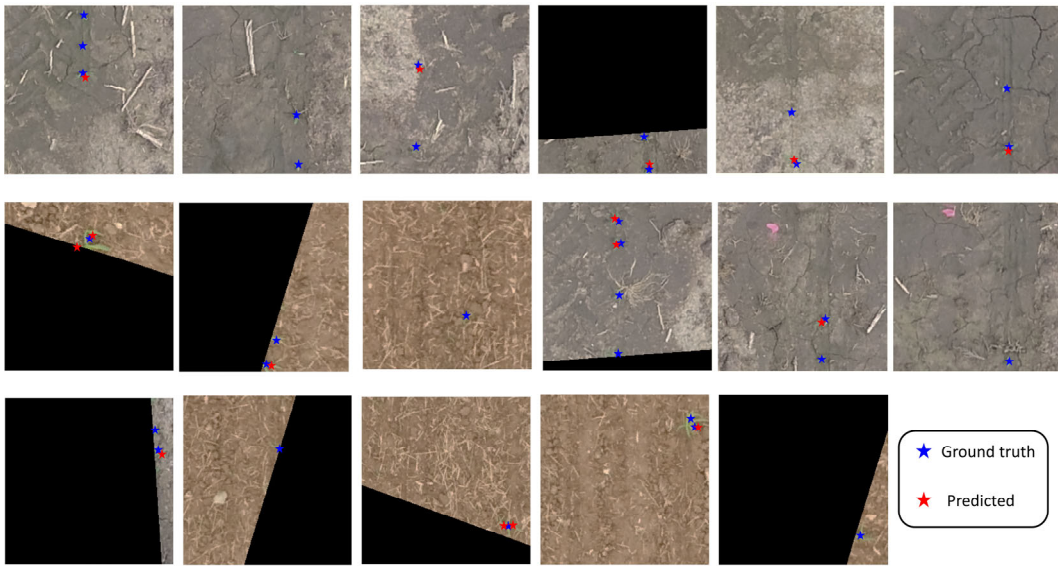


Figure 15. Extracted patches with normalized counting errors higher than 0.5.

4. Discussion

4.1. In-Depth Evaluation of the Results

The estimated AHDs of all patches in both datasets are shown in Figure 16 to analyze the proposed method's capability in plant localization in more detail. It can be observed that in some cases, the AHD value is much higher than the normal AHD, which is due to the omission or commission errors. The AHD would be even higher if both of these errors happened in a patch. For instance, according to Figure 13, patch #105 in the first dataset has zero counting errors. However, referring to Figure 16a, the AHD is 79.81 pixels, which is the highest among the other patches.

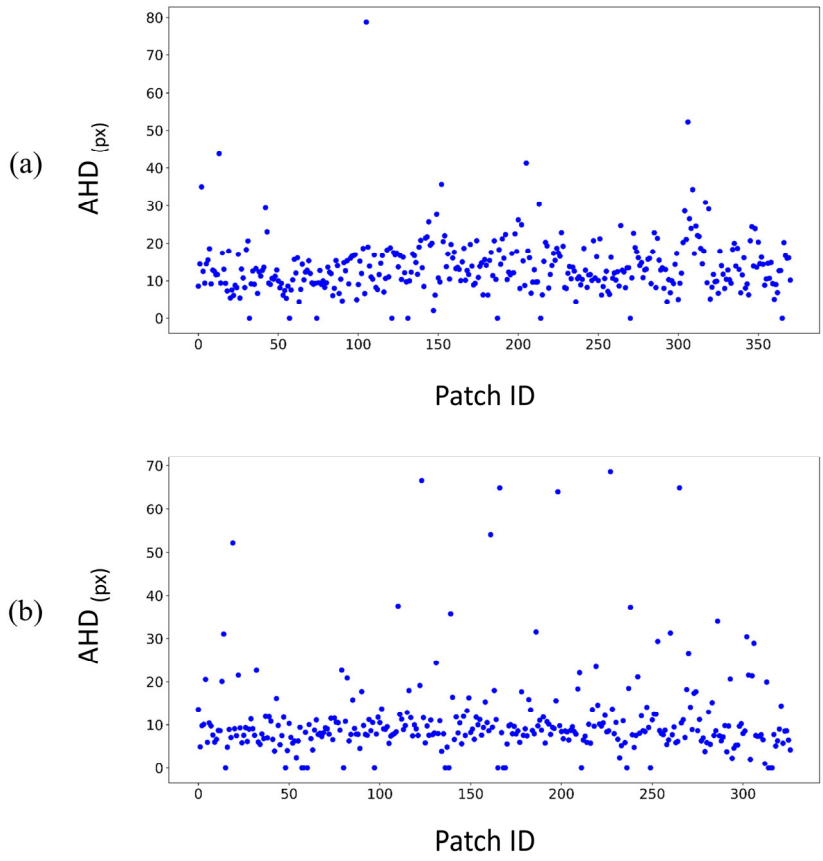


Figure 16. AHD values for each image patch in the first **(a)** and second datasets (**b**).

Figure 17 compares the ground truth and the predicted results of a sample patch with both omission and commission errors. It can be seen that the object-detector has made a mistake in detecting right-side objects; however, the number of detected targets is correct. As a result, counting error metrics such as the MAE may not be a reliable metric for evaluating the object-counting methods' robustness, although they are commonly used in the literature. Therefore, we tried to evaluate the proposed method's detection capability based on generating a binary image from ground truth and predicted objects' centroids.

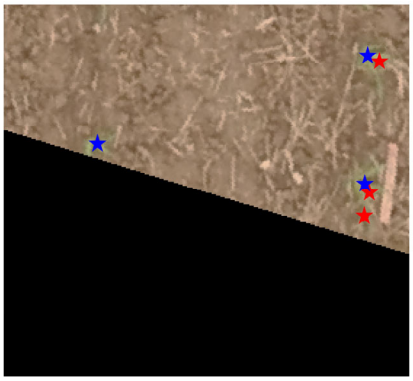


Figure 17. The image patch with high AHD and zero counting errors. Blue stars are ground truth points, and the red ones are predicted based on the model.

To this end, we represented each object as a circle with a specific radius. The reason is that our generated ground truth file is based on the center coordinate of objects (not the objects' bounding

box). Increasing the radius may increase the common areas between the ground truth and predicted map and, consequently, increase the detection quality. Table 5 shows the detection results based on the mean precision, recall, and F1 score metrics for different radius values. However, our proposed method automatically detected and localized plants, with more than 70% mean precision and recall for both datasets.

Table 5. Plant detection evaluation results based on generating a binary detection map for investigated image scenes.

Radius	Dataset	Mean Precision (×100)	Mean Recall (×100)	Mean F1 (×100)
R = 5 px	Dataset 1	71.49	71.14	71.20
	Dataset 2	72.63	71.69	71.76
R = 10 px	Dataset 1	73.76	73.42	73.39
	Dataset 2	80.87	79.31	79.51
R = 15 px	Dataset 1	79.41	78.95	79.00
	Dataset 2	86.06	83.32	84.21
R = 20 px	Dataset 1	84.52	83.90	84.05
	Dataset 2	89.04	85.89	87.00

Table 6 compares the proposed method with other state-of-the-art counting methods based on artificial intelligence and computer vision techniques. These methods have mostly used more than 70% training data and several augmentation strategies to increase their models' generalization capability. However, our method can generate various training data with different situations, which may cause a generalized model with fewer but more diverse training data. Similar to the method of Osco et al. [25], our method splits each dataset into non-overlapping 256 × 256 small patches. However, this procedure's main difference is that this procedure is based on the HT algorithm to find plant lines. Therefore, compared to most state-of-the-art methods, the proposed method has the advantage of not requiring any training data, gathering of ground truth, or data augmentation. It generates its training data in an automated approach, which also distinguishes it from unsupervised approaches such as the method proposed by Sun et al. [49] for detecting and counting cotton balls.

Compared to other simulated learning methods, the proposed method generates training data based on the scene's extraction of real objects. For example, Rahnemoonfar and Sheppard [37] trained their model by simulating tomato visual features. On the other hand, the results presented in [22] had a better AHD compared to our results, which could be because of AHD-based optimization and more training data (80% percent of the whole dataset); however, this was at the cost of higher computational complexity. Notably, the MAE or ACC measures cannot be compared with these methods due to the difference in dataset sizes and plant density; our method resulted in promising MAE and ACC. Overall, our proposed method's accuracy is comparable or even, in some cases, better than other plant-counting methods.

It is also worth mentioning that the proposed method uses a three-channel RGB image and does not exploit the infrared (IR) spectral bands of multispectral sensors. However, near-IR (NIR) bands are widely used for detecting different plant species. For example, in [25], the authors used a multispectral sensor containing RGB–NIR bands for detecting citrus trees. Although using multispectral sensors is advantageous for detecting plant species, they prevent transfer learning for very deep models (e.g., ResNet-101). Therefore, an extensive training process is needed with customized deep models.

Table 6. Comparison between the proposed method and other state-of-the-art counting methods in precision farming (ACC: Accuracy; RMSE: Root mean squared error; MAE: Mean absolute error; AHD: Average Hausdorff distance; FCN: Fully convolutional network; CNN: Convolutional neural network; MLP: Multi layer perceptron; UAV: Unmanned aerial vehicle).

Study	Object Detector	Dataset	Training	Counting Results
[37]	Modified Inception-ResNet	100 real images from Google	24,000 simulated images	ACC = 0.9103 RMSE = 2.52
[22]	FCN	5000 patches from UAV scene	80% training (5000 patches)	ACC: 0.958 MAE: 1.9 AHD: 7.1px (0.75 cm)
[22]	Faster R-CNN	5000 patches from UAV scene	80% training (5000 patches)	ACC: 0.823 MAE: 9.4 AHD: 9.0 px (0.75 cm)
[21]	Two FCNs	40 UAV scenes, each scene more than 10,000 plants	900 patches of size 512 × 512	ACC: 0.8194 MAE: 1600/scene
[50]	Feature fusion of MLP networks	128 images	50% training	MAE = 0.48
[51]	CNN + MLP	800 images	80% training	MAE = 0.83
[49]	Image processing	210 images	Automatic	ACC = 0.846 RMSE = 7.4
[24]	FCN encoder–decoder based on ResNet-101	Camera images	3000 image patches of size 300 × 300	RMSE = 1.69~3.4
[25]	FCN	Multispectral UAV	80% training (448 patches of 256 × 256)	MAE = 2.05 RMSE = 2.96
Our Method	**Faster R-CNN**	**UAV scene**	**800 simulated patches of size 256 × 256**	**ACC: 0.89~0.92 MAE: 0.25~0.57 AHD: 10.98~11.51**

4.2. Challenges and Future Works

The proposed method uses the ExG as an indicator of greenness and the HT to find the crop rows. Therefore, it depends on the plants' color and size, their discriminability from the background, and their abundance in the image. However, most of the fields follow a similar planting system, with wet or dry soil in the image background. Furthermore, objects with mostly different colors can be discriminated from the background by spectral indices. For example, Sun et al. [49] could discriminate cotton bolls from a soil background in an unsupervised manner, although cotton bolls are white and not green.

The developed automated image parsing method strongly depends on the efficiency of the crop row estimation. Even though the HT is a reliable and robust line detector, it may fail in places with dense plant distribution. Therefore, in this case, the crop rows have to be determined manually or based on a small part of the user's image. In addition, the image must contain only one field with uniform crop rows, and image scenes with various farms or crop rows should be divided into smaller scenes.

This paper used the pre-trained weights of the COCO dataset for faster convergence of the training process. Therefore, the proposed framework is only useful for RGB images, and in the case of using multispectral data, the pre-trained weights must be ignored. However, other object detection methods such as fully convolutional network (FCN) architectures would be another option.

The proposed framework was tested on single object detection, a unique type of plant and crop (cornfield), but with different planting, soil, and illumination conditions. Therefore, optimizing the framework for other crop types and fields is required. It is also necessary to consider the image resolution and the plants' density compared to the background to optimize the patch size and data simulation conditions; however, they follow the proposed algorithm's same strategy.

Therefore, as future work, we may evaluate the framework's multi-class detection capability and develop a new deep architecture with better performance in a more complex and challenging environment, e.g., extracting and counting dense trees. FCN has been extensively used for segmentation or counting in crowded areas. However, more research is needed for developing unsupervised and automated frameworks based on FCNs for the object (plant) counting and localization.

5. Conclusions

This paper proposed an automated framework for detecting and counting plants in RGB images captured by drones. This framework simulated the training patches by dividing the soil background and plants using the ExG index and C-means binary clustering and evaluating the statistics of the extracted objects' areas. Additionally, the HT algorithm was used to detect the crop rows to rotate the image in the direction of the vertical, or horizontal, crop rows to prepare the image scene for plant detection. The faster R-CNN algorithm if the core of this framework, which is trained by simulated data to detect plants from drone images.

The applicability of the proposed framework was evaluated using two different drone images from two cornfields. The first and second datasets were divided into 442 and 328 image patches, respectively. According to the counting metrics, the proposed method could detect and count 92.4% and 89.4% of the objects with a mean absolute error of 0.253 plants per patch in the first dataset and 0.571 plants per patch in the second dataset. The AHD, mean precision, recall, and F1 score metrics were utilized to evaluate the object localization and detection results. The acquired localization errors for the first and second datasets, based on the AHD metric, were 11.51 and 10.98 pixels. Additionally, the mean precision, recall, and F1 score results were 0.845, 0.890, and 0.839 for the first dataset, and 0.859, 0.841, and 0.870 for the second one, in a radius of 20 pixels.

Compared to other recent methods for counting and detecting plant canopy in PA, our proposed automated method led to promising results. An unsupervised and automated method for generating training data to be employed in the faster R-CNN framework is advantageous to the proposed algorithm. Furthermore, using the ExG index and C-means clustering algorithm was helpful for initial plant detection. Overall, the obtained results proved the reliability of the proposed framework for plant detection and counting.

Author Contributions: B.H. contributed to implementing the proposed methodology, analyzing the results, and writing the first draft of the manuscript. H.R. designed and directed the research project and methodology, contributed to the implementation, analyzed the results, and reviewed the manuscript. S.H. contributed to the analysis of the results and reviewing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Godfray, H.C.J.; Beddington, J.R.; Crute, I.R.; Haddad, L.; Lawrence, D.; Muir, J.F.; Pretty, J.; Robinson, S.; Thomas, S.M.; Toulmin, C. Food security: The challenge of feeding 9 billion people. *Science* **2010**, *327*, 812–818.
2. Gikunda, P.; Jouandeau, N. State-Of-The-Art Convolutional Neural Networks for Smart Farms: A Review. In Proceedings of the Science and Information (SAI) Conference, London, UK, 18–20 July 2017.
3. Seelan, S.K.; Laguette, S.; Casady, G.M.; Seielstad, G.A. Remote sensing applications for precision agriculture: A learning community approach. *Remote Sens. Environ.* **2003**, *88*, 157–169, doi:10.1016/j.rse.2003.04.007.
4. Kerkech, M.; Hafiane, A.; Canals, R. Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Comput. Electron. Agric.* **2018**, *155*, 237–243, doi:10.1016/j.compag.2018.10.006.
5. Mulla, D.J. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosyst. Eng.* **2013**, *114*, 358–371.
6. Servadio, P.; Verotti, M. Fuzzy clustering algorithm to identify the effects of some soil parameters on mechanical aspects of soil and wheat yield. *Span. J. Agric. Res.* **2018**, *16*, 5.
7. Zhao, H.; Yuan, Q.; Song, S.; Ding, J.; Lin, C.-L.; Liang, D.; Zhang, M. Use of Unmanned Aerial Vehicle Imagery and Deep Learning UNet to Extract Rice Lodging. *Sensors* **2019**, *19*, 3859, doi:10.3390/s19183859.
8. Sankey, T.; Donager, J.; McVay, J.; Sankey, J.B. UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA. *Remote Sens. Environ.* **2017**, *195*, 30–43.
9. Lottes, P.; Khanna, R.; Pfeifer, J.; Siegwart, R.; Stachniss, C. UAV-based crop and weed classification for smart farming. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017.
10. Xiang, H.; Tian, L. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosyst. Eng.* **2011**, *108*, 174–190.
11. Jin, X.; Liu, S.; Baret, F.; Hemerlé, M.; Comar, A. Estimates of plant density of wheat crops at emergence from very low altitude UAV imagery. *Remote Sens. Environ.* **2017**, *198*, 105–114.
12. Walter, A.; Khanna, R.; Lottes, P.; Stachniss, C.; Siegwart, R.; Nieto, J.; Liebisch, F. Flourish-a robotic approach for automation in crop management. In Proceedings of the International Conference on Precision Agriculture (ICPA), Montreal, QC, Canada, 24–27 June 2018.
13. Mukherjee, A.; Misra, S.; Raghuwanshi, N.S. A survey of unmanned aerial sensing solutions in precision agriculture. *J. Netw. Comput. Appl.* **2019**, *148*, 102461.
14. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36, doi:10.1109/MGRS.2017.2762307.
15. Gong, M.; Yang, H.; Zhang, P. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. *ISPRS J. Photogramm. Remote Sens.* **2017**, *129*, 212–225, doi:10.1016/J.ISPRSJPRS.2017.05.001.
16. Hosseiny, B.; Rastiveis, H.; Daneshtalab, S. Hyperspectral image classification by exploiting convolutional neural networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 535–540.
17. Hosseiny, B.; Shah-Hosseini, R. A hyperspectral anomaly detection framework based on segmentation and convolutional neural network algorithms. *Int. J. Remote Sens.* **2020**, *41*, 6946–6975.
18. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90.
19. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning—Method overview and review of use for fruit detection and yield estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234.
20. Dijkstra, K.; van de Loosdrecht, J.; Schomaker, L.R.B.; Wiering, M.A. Centroidnet: A deep neural network for joint object localization and counting. In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2018*; Lecture Notes in Computer Science; Springer: Dublin, Ireland, 2018; Volume 11053 LNAI, pp. 585–601.
21. Wu, J.; Yang, G.; Yang, X.; Xu, B.; Han, L.; Zhu, Y. Automatic counting of in situ rice seedlings from UAV images based on a deep fully convolutional neural network. *Remote Sens.* **2019**, *11*, 691.
22. Ribera, J.; Guera, D.; Chen, Y.; Delp, E.J. Locating objects without bounding boxes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6479–6489.

23. Attouch, H.; Lucchetti, R.; Wets, R.J.-B. The topology of theϱ-hausdorff distance. *Annali di Matematica Pura ed Applicata* **1991**, *160*, 303–320.
24. Bellocchio, E.; Ciarfuglia, T.A.; Costante, G.; Valigi, P. Weakly Supervised Fruit Counting for Yield Estimation Using Spatial Consistency. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2348–2355.
25. Osco, L.P.; de Arruda, M.d.S.; Marcato Junior, J.; da Silva, N.B.; Ramos, A.P.M.; Moryia, É.A.S.; Imai, N.N.; Pereira, D.R.; Creste, J.E.; Matsubara, E.T.; et al. A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 97–106, doi:10.1016/j.isprsjprs.2019.12.010.
26. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
27. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; 2015. pp. 91–99. Available online: papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks (accessed on 26 October 2020).
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
31. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
32. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors* **2016**, *16*, 1222, doi:10.3390/s16081222.
36. Zhou, C.; Ye, H.; Hu, J.; Shi, X.; Hua, S.; Yue, J.; Xu, Z.; Yang, G. Automated Counting of Rice Panicle by Applying Deep Learning Model to Images from Unmanned Aerial Vehicle Platform. *Sensors* **2019**, *19*, 3106.
37. Rahnemoonfar, M.; Sheppard, C. Deep count: Fruit counting based on deep simulated learning. *Sensors* **2017**, *17*, 905, doi:10.3390/s17040905.
38. Bah, M.; Hafiane, A.; Canals, R. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. *Remote Sens.* **2018**, *10*, 1690, doi:10.3390/rs10111690.
39. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*; Springer: Zurich, Switzerland, 2014; pp. 740–755.
40. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: New York, NY, USA, 2009; pp. 248–255.
41. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368.
42. Wachowiak, M.P.; Walters, D.F.; Kovacs, J.M.; Wachowiak-Smolíková, R.; James, A.L. Visual analytics and remote sensing imagery to support community-based research for precision agriculture in emerging areas. *Comput. Electron. Agric.* **2017**, *143*, 149–164.
43. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*; Academic Press: Cambridge, MA, USA, 2009; ISBN 9780080949123.
44. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 0262337371.

45. Rahnemoonfar, M.; Dobbs, D.; Yari, M.; Starek, M.J. DisCountNet: Discriminating and counting network for real-time counting and localization of sparse objects in high-resolution UAV imagery. *Remote Sens.* **2019**, *11*, 1128.
46. Duda, R.O.; Hart, P.E. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **1972**, *15*, 11–15.
47. Rastiveis, H.; Shams, A.; Sarasua, W.A.; Li, J. Automated extraction of lane markings from mobile LiDAR point clouds based on fuzzy inference. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 149–166, doi:10.1016/j.isprsjprs.2019.12.009.
48. Staging Corn Growth | Pioneer Seeds Available online: https://www.pioneer.com/us/agronomy/staging_corn_growth.html (accessed on 18 October 2020).
49. Sun, S.; Li, C.; Paterson, A.H.; Chee, P.W.; Robertson, J.S. Image processing algorithms for infield single cotton boll counting and yield prediction. *Comput. Electron. Agric.* **2019**, *166*, 104976.
50. Giuffrida, M.V.; Doerner, P.; Tsaftaris, S.A. Pheno-Deep Counter: A unified and versatile deep learning architecture for leaf counting. *Plant J.* **2018**, *96*, 880–890.
51. Itzhaky, Y.; Farjon, G.; Khoroshevsky, F.; Shpigler, A.; Bar-Hillel, A. Leaf counting: Multiple scale regression and detection using deep CNNs. In Proceedings of the BMVC, Newcastle, UK, 3–6 September 2018; p. 328.