



Article

Advancing Tassel Detection and Counting: Annotation and Algorithms

Azam Karami ^{1,*} , Karoll Quijano ² and Melba Crawford ^{1,3}¹ Department of Agronomy, Purdue University, West Lafayette, IN 47907, USA; mcrawford@purdue.edu² Department of Environmental and Ecological Engineering, Purdue University, West Lafayette, IN 47907, USA; kquijano@purdue.edu³ School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA

* Correspondence: akaramir@purdue.edu; Tel.: +1-765-496-9355

Abstract: Tassel counts provide valuable information related to flowering and yield prediction in maize, but are expensive and time-consuming to acquire via traditional manual approaches. High-resolution RGB imagery acquired by unmanned aerial vehicles (UAVs), coupled with advanced machine learning approaches, including deep learning (DL), provides a new capability for monitoring flowering. In this article, three state-of-the-art DL techniques, CenterNet based on point annotation, task-aware spatial disentanglement (TSD), and detecting objects with recursive feature pyramids and switchable atrous convolution (DetectoRS) based on bounding box annotation, are modified to improve their performance for this application and evaluated for tassel detection relative to Tasselnetv2+. The dataset for the experiments is comprised of RGB images of maize tassels from plant breeding experiments, which vary in size, complexity, and overlap. Results show that the point annotations are more accurate and simpler to acquire than the bounding boxes, and bounding box-based approaches are more sensitive to the size of the bounding boxes and background than point-based approaches. Overall, CenterNet has high accuracy in comparison to the other techniques, but DetectoRS can better detect early-stage tassels. The results for these experiments were more robust than Tasselnetv2+, which is sensitive to the number of tassels in the image.

Keywords: maize; tassel detection; deep learning; annotation

Citation: Karami, A.; Quijano, K.; Crawford, M. Advancing Tassel Detection and Counting: Annotation and Algorithms. *Remote Sens.* **2021**, *13*, 2881. <https://doi.org/10.3390/rs13152881>

Academic Editors: Mercedes E. Paoletti and Juan M. Haut

Received: 30 May 2021
Accepted: 8 July 2021
Published: 23 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Maize is a major crop for food consumption and a source of material for a wide range of products. Increasing maize yield is important, especially under the pressure of global climate change, which is often associated with increased temperatures and extreme droughts [1,2]. Plant breeders focused on developing improved varieties of crops seek to understand the joint impact of genetics, environment, and management practices on yield. A key component of their programs involves measuring various physical, chemical, and biological attributes of the plants, referred to as phenotypes, throughout the growing season. For maize and many other grains crops, flowering is one of the most important stages as it initiates the stage of reproduction. Any external stress, physical or biological, can cause plant damage and result in production losses. Traditional approaches for field-based monitoring of tasseling are manual and thus time-consuming, labor-intensive, expensive, and potentially error-prone, especially in large fields. Alternatively, image-based techniques that automatically detect and count the tassels to predict the flowering date can mitigate these problems. However, varying illumination and shape, shadows, occlusions, and complex backgrounds impact the accuracy of these approaches [3,4].

Recently, remote sensing (RS) imagery acquired by UAVs has been investigated for counting objects such as plants, because high temporal and spatial resolution data can be acquired over large fields [5]. In this study, data acquired by an RGB camera mounted on a UAV are investigated for tassel detection and counting. Manual tassel counting in the large

orthophotos is similar to field-based manual counting in the sense of being a subjective, tedious task and should be performed by experts because of the highly overlapped and complex shape of tassels as they develop. Various machine learning (ML) approaches have been introduced to address these problems, with some achieving high accuracy, along with a cost reduction [6]. These approaches are mainly categorized as classic and DL methods. Classic approaches, such as support vector machines (SVMs) and morphological operators, are very sensitive to the background, illumination, soil color, shadow, shape, and size of tassels [1,7–9]. For example, the tassel image-based phenotyping system (TIPS) is a platform for extracting tassels from plants. TIPS removes the background noise from RGB images, then creates binary images and smooths and skeletonizes the tassel [7]. A combination of the histogram of oriented gradients and SVM is used to detect the tassels. Ye et al. [8] consider spatiotemporal saliency mapping to decrease the background effect and increase the tassel detection accuracy. In [9], an algorithm based on color transformation, SVM, and morphological operators is used for maize tassel counting. An Itti saliency-based visual attention system is applied to detect the potential pixels of maize tassels, and a color attenuation prior is applied to remove image saturation and decrease the effects of sunlight [1]. DL models have also demonstrated excellent performance for many precision agriculture applications [10] such as plant counting [11,12], wheat spike detection [13], sorghum head detection [14], and leaf segmentation [15]. Recently, two well-known DL strategies have been investigated for tassel counting [16]:

(1) Regression-based techniques: These methods can only count, as they regress the local count calculated from the density map, usually estimating non-integer counts. More information about the tassel's location and number—for example, the number of true positives (TP), false positives (FP), and false negatives (FN)—cannot be determined. However, these techniques are faster than detection-based approaches. TasselNetv2+, which was introduced to count tassels based on regression CNN [17], has undergone multiple improvements through subsequent implementations. Visual context was added to the local patches in the CNN in TasselNetv2 [18], and the first layer of the CNN was modified with global average pooling and implemented using PyTorch in TasselNetv2+ [19]. All three implementations use point annotation.

(2) Detection-based techniques: These approaches are categorized as anchor-based and anchor-free. Anchor-based approaches include one- and two-stage detectors and are based on bounding box annotation. Single-stage detectors consider object detection as a dense classification and localization problem [20–23]. They are faster and simpler, but detection accuracy is usually lower than two- or multi-stage detectors. Two-stage detectors first generate the object proposals, and then, in the second stage, the features are extracted from the candidate proposals [24–27]. These detectors have high localization and object detection accuracy. Anchor-based approaches that have been used for maize tassel detection include Faster R-CNN [24,28], Yolov3 [29], RetinaNet [4,20], and FaceBoxes [30]. Of these, Faster R-CNN obtained the highest accuracy [16,28,31]. Anchor-free detectors do not generate the anchors; therefore, the computational complexity is typically decreased. These approaches are mainly anchor-point methods (e.g., Grid R-CNN [32] and FoveaBox [33]) and key-point detectors (e.g., CornerNet [34], ExtremeNet [35], and CenterNet [36]). These techniques, which have been successfully demonstrated for other applications with similar complexity, could potentially yield higher accuracy tassel detection and counting than previously published approaches without the excessive computational overhead. To the best of our knowledge, these approaches have not been investigated for this application.

The performance of DL approaches is highly dependent on the quality of the annotated dataset which is used for training and validation. Tassels shown in imagery are highly overlapped and complex in shape and differ in size. DL architectures are typically implemented with specific forms of annotation. In this study, two common types of annotation, point and bounding boxes, were investigated in conjunction with three DL architectures. We also modified all three approaches to potentially improve their performance for the tassel detection and counting problem. The anchor-free CenterNet architecture in [5,36] uses the

center point of each tassel for detection, which is simpler to implement than bounding box annotation. The size and dimension of the objects are directly calculated without any prior anchor. The CenterNet Hourglass network was modified and implemented with few-shot learning for this study. Two anchor-based, multi-stage detectors based on the bounding box annotations [25,27], which obtained high accuracy on the COCO dataset, were also modified and implemented for tassel detection. Specifically, the loss function for TSD was modified during the training process, and the classification and regression problems were considered separately, which increased the tassel detection accuracy [25]. In DetectoRS, the existing feature pyramid networks (FPN) were modified, and extra feedback connections were added to the backbone. The convolution layers of ResNext were also replaced with the atrous convolutions and deformable convolutional networks (DCNs) [27].

The remainder of this paper is organized as follows. In Section 2, the study area and details of point and box annotations are described. The three state-of-the-art DL-based algorithms are also introduced, and the specific modifications are described for improved tassel detection using multiple evaluation metrics. Experimental results for the multiple aspects of the study are presented and discussed in Section 3. Section 4 provides a critical evaluation of the approaches and discusses directions for future research.

2. Materials and Methods

2.1. Field Experiment and Image Acquisition

The experiment was carried out at the Agronomy Center for Research and Education (ACRE) of Purdue University (40°28'43" N, 86°59'23" W, 4540 US-52, West Lafayette, IN, USA) during the 2020 growing season (see Figure 1). The field experiment was planted in a modified randomized complete block design (RCBD), using varieties of maize from the Genomes to Fields (G2F) initiative for High-Intensity Phenotype Sites (G2F-HIPS). Two replications of 22 entries for hybrids (G2F-HIPS) and 22 entries for inbreds (G2F-HIPS) were planted on 12 May in a two-row segment plot layout with a plant population of 30,000 plants per acre.

A total of 88 plots were imaged on 20 July using a Sony Alpha 7R-III RGB camera mounted on a UAV DJI Matrice M600 Pro, with a Trimble APX-15v3 GNSS/INS unit for direct georeferencing. The UAV was flown at 20 m altitude, and the RGB imagery was processed to a 0.25 cm pixel resolution orthophoto using the method of [37] to eliminate inaccuracies due to lens distortion and double mapping associated with significant height changes over short distances. In-field manual phenotyping data were recorded throughout the season as stipulated by the G2F-HIPS standard operating procedures [38], including stand count and anthesis date. During the flowering period, visual inspections were performed to determine the anthesis date of each plot. Hybrids and inbreds in this experiment had different anthesis dates, with a range of 20 days from the first variety to flower to the last. This provided an opportunity to evaluate the counting algorithms over a range of flowering times in the same field.

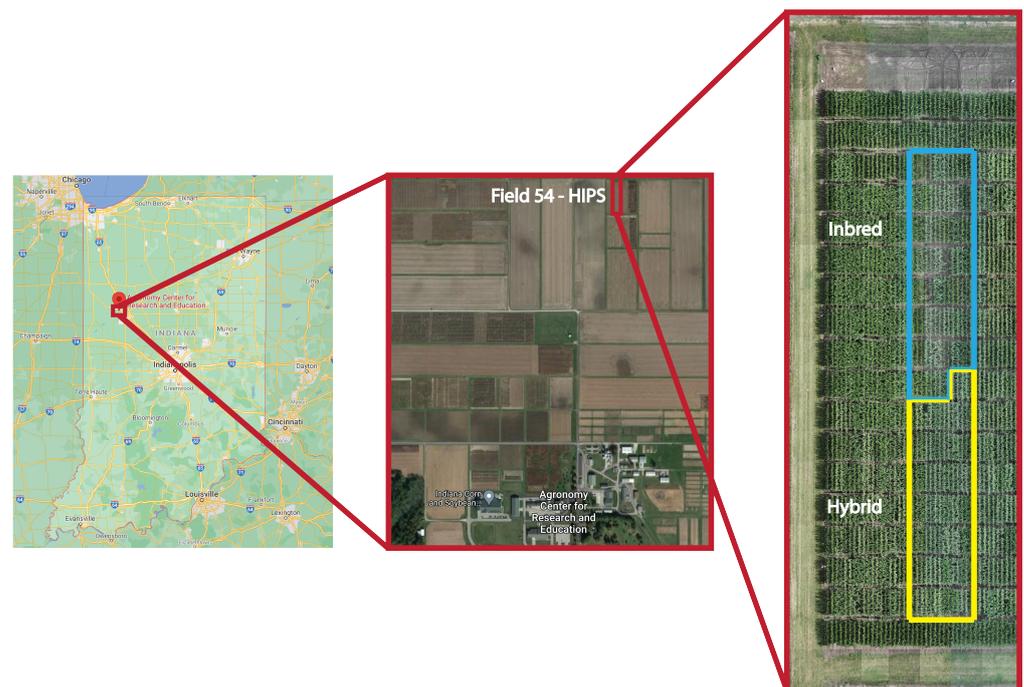


Figure 1. Experimental site: the field trials at Agronomy Center for Research and Education (ACRE).

2.2. Data Annotation

A rigorous tassel annotation was performed for this study using the open source annotation tool LabelMe [39]. First, the tassels are annotated in the orthophoto with points at the center of the tassel (see Figure 2a), designating the position of the tassel and the number of tassels per row within the plot. Multiple reviews are needed, as mistakes are common, even for experienced labelers. As different algorithms require specific forms of annotation as inputs, a second annotation dataset was developed from the point annotations, where bounding boxes of 20×20 pixels were generated from the previously annotated points (see Figure 2b). The third annotation dataset (see Figure 2c) was developed using bounding boxes where the size of the boxes was manually adjusted to the size of the tassel, avoiding excessive overlap with the neighboring boxes. The impact of the annotation approaches is evaluated in the Results section.

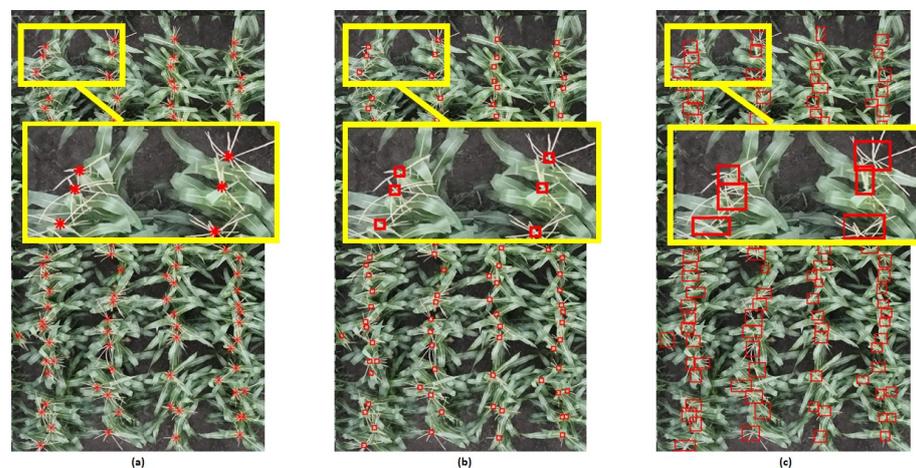


Figure 2. Tassel annotation: (a) Point; (b) Bounding box; (c) Modified bounding box.

2.3. Model Description

2.3.1. CenterNet

CenterNet, a state-of-the-art anchor-free detector [36], which was recently demonstrated to be effective for plant counting, was investigated for the tassel counting problem because of the simplicity of the annotation and limited computational requirements [5]. For tassel detection and counting, the point annotation dataset described in the previous section was employed for the training process. For localization, CenterNet uses a Gaussian kernel and a fully connected network (FCN) to create the heatmap, which is used to estimate the tassel centers. CenterNet does not require any post-processing, which reduces computational complexity. For this study, the CenterNet-based approach with an Hourglass-104 architecture was implemented to determine the locations of tassels' centers (see Figure 3 for more details). The hyperparameters such as learning rate, number of epochs, and batch size were optimized by grid search for tassel counting.

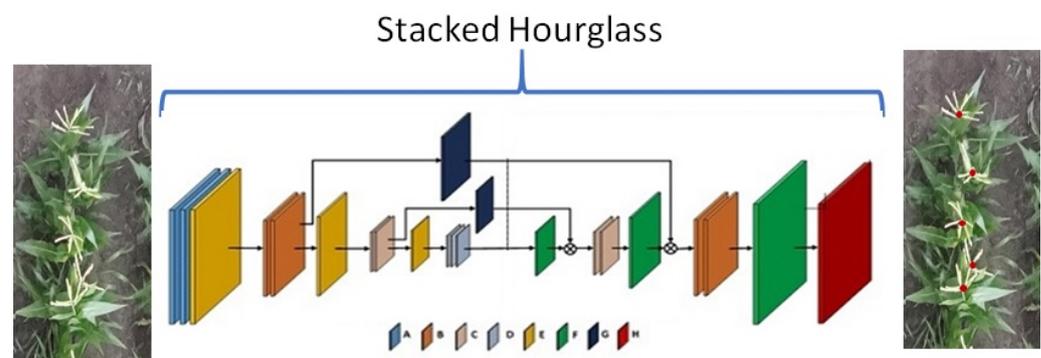


Figure 3. CenterNet Schematic “Hourglass-104 architecture and illustration of the heatmap for the input image. A and B are convolutional layers; C and D are inception modules; E is the max-pooling layer; F is the transposed convolutional layer; G is the residuals modules; H is the loss layer”.

2.3.2. TSD

In TSD [25], the spatial misalignment between classification and regression functions in the sibling’s head can decrease the object detection accuracy. These functions are decoupled by creating two spatially disentangled proposals. As shown in Figure 4, the original images are input to the backbone, and a regional proposal P is then generated by the region proposal network (RPN). In the next step, two separate proposals, \hat{P}_c and \hat{P}_r , are estimated for classification and regression. Finally, the object is detected, and the coordinated box is regressed.

Three modifications of TSD were implemented for tassel detection, as the shape and size of the tassels are highly variable from one variety to another: (1) Cascade R-CNN, a multi-stage extension of two-stage R-CNN, was used instead of Faster R-CNN. The architecture includes a sequence of detectors, where the output of one stage is used to train the next stage. This improves the threshold of intersection over union (IoU) metric compared to sequential detectors. The value is gradually increased without overfitting, and the number of false positives is reduced. For inference, consecutive detectors can also significantly increase the detection accuracy [40]. (2) DCNs were added to the backbone, replacing CNNs which have fixed geometric structures and are not appropriate for tassel detection. This is also due to the limited amount of training data. For detecting rotated or scaled objects, the training images are augmented, and different augmentation transforms are considered to achieve reasonable detection accuracy [41]. (3) To address the variation in the size of the tassels, a multiscale test was also implemented.

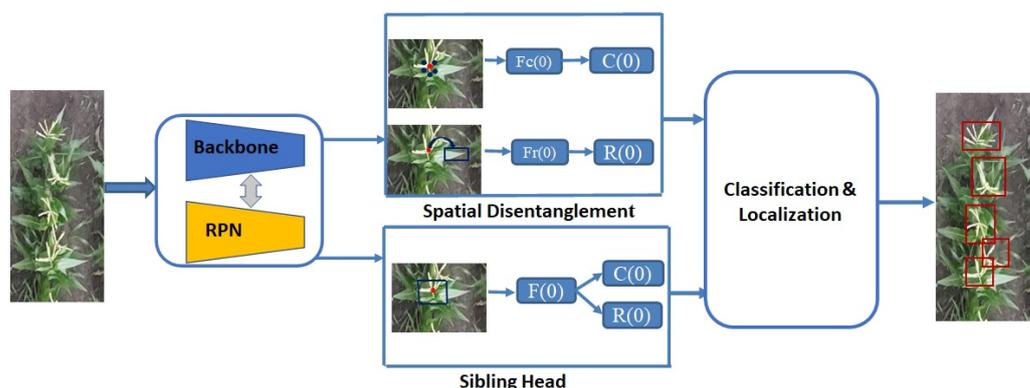


Figure 4. TSD Schematic (ResNext-101).

2.3.3. DetectoRS

DetectoRS obtained the highest accuracy on the COCO dataset in 2020 [27]. It uses twice looking and thinking ideas, following the architecture of Cascade-RCNN, and implements subsequent detectors. DetectoRS includes two main steps: recursive feature pyramids (RFP) and switchable atrous convolution (SAC). The existing feature pyramid networks (FPN) were modified in DetectoRS by adding feedback connections from the FPN layers into the bottom-up backbone layers (see Figure 5), which help extract stronger features. In this study, the RFP and SAC were modified for tassel detection because of their variation in size and geometric complexity; details are explained in the following.

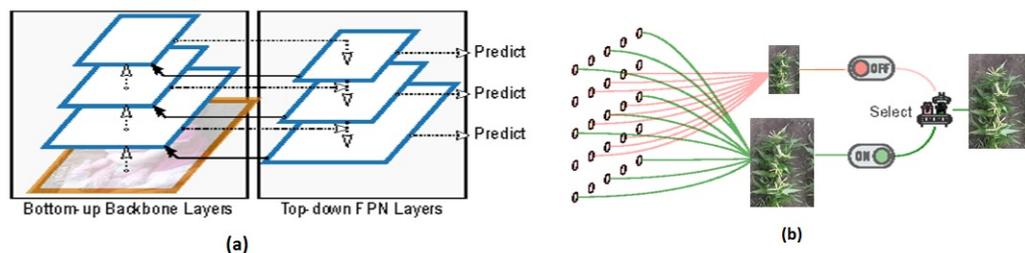


Figure 5. DetectoRS Schematic (ResNext-101); (a) RFP; (b) SAC.

In the original version of DetecoRS, two sequential RFPs are used. For the tassel detection, because of the variety in shape and size of tassels, three RFPs are considered (see Figure 6). Features at each stage are recursively extracted as follows:

$$f_i^t = F_i^t(f_{i+1}^t, x_i^t) \quad x_i^t = B_i^t(x_{i-1}^t, R_i^t(f_i^{t-1})) \quad (i = 1, 2, 3, t = 1, 2, 3) \quad (1)$$

where t : iteration number of RFP, i : number of decomposition level at each RFP, $f_i^0 = 0$, B_i^t : i -th stage of the bottom-up backbone at iteration t , F_i^t : i -th stage of the top-bottom at iteration t , R_i^t : shared function at iteration t , f_i^t : i -th output feature at iteration t .

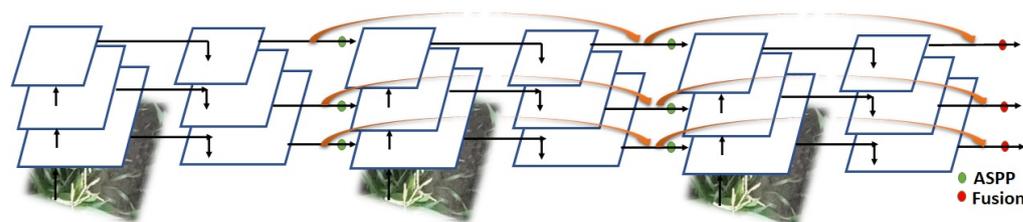


Figure 6. Tassel RFP.

In the original DetectoRS, the ResNet architecture is modified. ResNet has four similar stages; DetectoRS only changes the first layer by replacing it with a convolution layer

(kernel size one). This is called atrous spatial pyramid pooling (ASPP). Atrous convolutions add zeros to the original kernel and increase the size of the kernel, but the computational complexity is not increased [42,43]. The initial weight of this layer is set to zero, so the pre-trained weights from the ImageNet or COCO datasets can be used. For tassel detection, the ResNext architecture is used and modified. In the ResNext architecture, unlike ResNet, the neurons of one path are not connected to the neurons of other paths. It also uses the bottleneck design at each convolution path and reduces the computational complexity. ResNext has higher object detection accuracy than the ResNet backbone on the ImageNet data [44]. Additionally, DCNs have been used instead of CNN in the tassel ASPP (see Figure 7) because of variation in tassels' shapes and geometry.

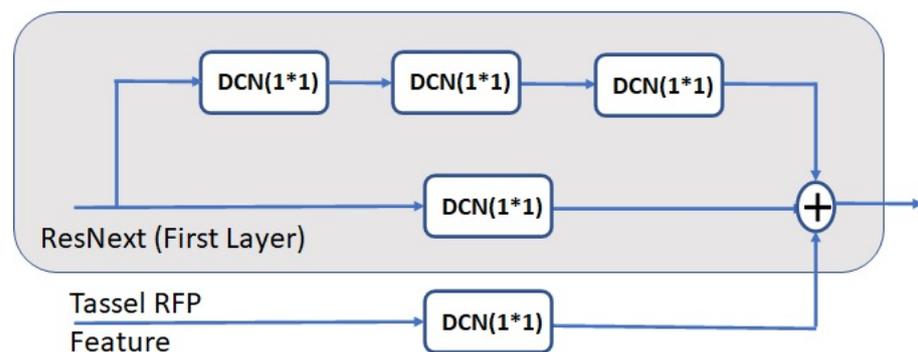


Figure 7. Tassel ASPP.

In Figure 5, SAC with the atrous rates 1 (red) and 2 (green) is shown; the same object at different scales can be easily detected using different atrous rates. Therefore, considering different atrous rates can increase the accuracy of tassel detection which has different sizes and shapes. For tassel detection, DCN with kernel size (1×1) is used (See Figure 8). The tassel SAC is calculated as follows:

$$DCN(x, w, r) \xrightarrow[TasselSAC]{Convert} S(x)DCN(x, w, r) + (1 - S(x))DCN(x, w + \Delta w, r) \quad (2)$$

where r is a hyperparameter, and for tassel detection, the optimal value is 5. Δw is a trainable parameter. The switchable function $S(\cdot)$ includes an average pooling with kernel size (5×5) followed by a convolution layer with size (1×1) . Based on the ideas in SENet [44], before and after SAC, DCN and global average pooling were added to increase the tassel detection accuracy. The multiscale test was also considered because of the variation in the size and shape of the tassels.

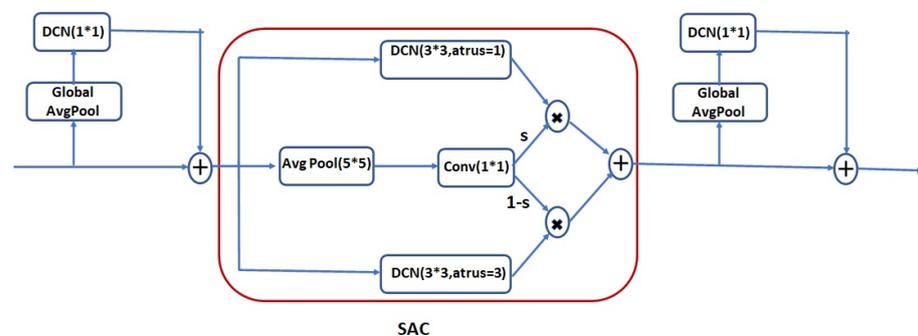


Figure 8. Tassel SAC.

2.3.4. TasselNetv2+

For maize tassel counting, the L_1 loss function which is used in the regression problem is calculated as follows:

$$L_1 = \frac{1}{m} \sum_{i=1}^m |a_i|_1 \quad (3)$$

where m is the number of training images, and a_i is the residual that measures the difference between the regressed count and the ground truth count for the i -th image [19]. The optimization problem aims to minimize L_1 using the Adam technique. The optimal regressed count is the estimated number of tassels in the image, which is not usually an integer number.

2.4. Parameter Settings

The optimal parameter settings for the four algorithms are determined experimentally and are shown in Table 1.

Table 1. Parameter settings.

Parameter	CenterNet	TSD	DetectoRS	TasselNetv2+
Model	ExtremeNet	CascadeRCNN	CascadeRCNN	-
Backbone	Hourglass	ResNeXt+DCN	ResNeXt	-
Depth	104	101	101	-
Batch Size	11	2	2	16
Epochs	240	500	500	300
Optimizer	Adam	SGD	SGD	SGD
Learning Rate (lr)	1.25×10^{-4}	1.25×10^{-3}	1.25×10^{-3}	1.25×10^{-2}
Gaussian Kernel Parameter	-	-	-	6

2.5. Model Evaluation

Two types of metrics for detection and regression-based approaches were implemented and evaluated.

2.5.1. Detection Metrics

The performance of detected tassels using bounding boxes-based approaches (DetectoRS and TSD) is evaluated by the following metrics:

Intersection over union IoU , which measures the overlap between a predicted bounding box B and B^{gt} , the ground truth bounding box (see Equation (4)), is a widely used metric. If the values of B and the B^{gt} match exactly, the value of IoU is one. For the tassel detection, the acceptable value is set to 0.5.

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \quad (4)$$

If the value of IoU is higher than 0.5, they are selected as detected objects and compared to the ground reference. The correct and incorrect detected tassels are then represented as TP and FP respectively. The missing tassels are also considered as FN . The precision (Pr), recall (Re), and score (Sc) values are calculated as:

$$Pr = \frac{TP}{TP + FP} \quad (5)$$

$$Re = \frac{TP}{TP + FN} \quad (6)$$

$$Sc = 2 \times \frac{Re \times Pr}{Re + Pr} \quad (7)$$

The accuracy of the number of the detected tassels N_t and ground truth N_g is defined as follows:

$$Acc = \frac{N_t}{N_g} \quad (8)$$

For the CenterNet based point annotation, the value of r is considered as the maximum distance between the ground truth and the predicted tassel's center location for consideration as a correct or missing tassel detection (assumed to be 10 cm). The TP , FP , and FN values are calculated based on the criteria introduced in [5] for plant counting.

2.5.2. Counting Metrics

The mean absolute error (MAE) and root mean square error ($RMSE$) indicate the difference between N_t and N_g and are obtained as:

$$MAE = \frac{1}{M} \sum_{i=1}^M |N_{gi} - N_{ti}| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (N_{gi} - N_{ti})^2} \quad (10)$$

3. Results

The tassel detection algorithms were implemented on a machine with seven cores, one GPU (GTX 1080ti, 11 GB RAM), and 128 GB external RAM.

3.1. Comparison of Original and Developed Anchor and Anchor-Free Based Approaches for Tassel Detection

As mentioned previously, the original version of anchor-free CenterNet and anchor-based approaches (TSD and DetectoRS) were introduced for object detection of the COCO dataset. Because of the variation in size, shape, similarity to the leaves, and overlap, the modifications described in Section 2.3 were implemented for the tassel dataset. In [40], it was reported that the Cascade R-CNN had a higher object detection accuracy than Faster R-CNN and Mask R-CNN on the COCO dataset (8 & 6%), respectively. In tassel detection, the Cascade R-CNN improved the average detection accuracy by $\sim(2.3$ & $2.7\%)$ for TSD and DetectoRS, respectively, in comparison to Faster R-CNN. Mask R-CNN requires the mask annotation around the tassels and was not implemented because of the complexity in the tassel shape and their high overlap. The results in Figure 9 show that applying the modifications increased the mean tassel detection accuracy and reduced the corresponding standard deviation.

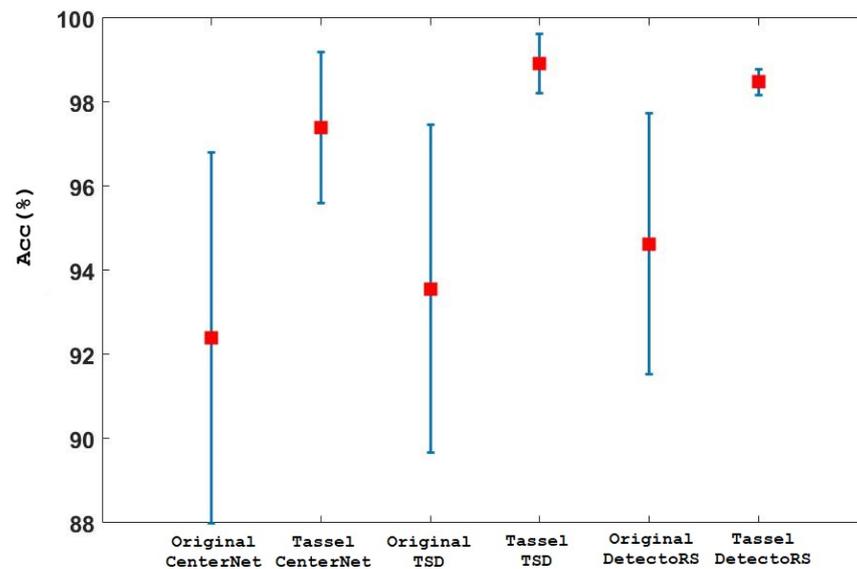


Figure 9. Comparison of original and developed tassel detection algorithms.

3.2. Sensitivity Analysis to Bounding Box Sizes

The bounding box-based approaches are sensitive to the size of the bounding boxes and their overlap for mature tassels. Figure 10 shows three different sizes of bounding boxes used in the TSD algorithm. In Figure 10a, the small bounding boxes were drawn around the tassels' centers originally obtained from the point annotation. These bounding boxes did not fully include the tassels and frequently missed the shape. Therefore, the detection accuracy was reduced. Figure 10b depicts the bounding boxes which encompass the tassel. In this image, and for most of the row segments where the tassels have fully flowered, the bounding boxes have highly overlapping areas. In Figure 10c, the bounding boxes were selected as large as possible to include the tassel area, while attempting to reduce the overlapping areas. This sensitivity to the size of the bounding box during annotation reduces the practicality of using it. The number of tassels in the ground truth is 61, and TSD detected 44 ($TP = 43$, $FP = 1$, $FN = 18$), 46 ($TP = 45$, $FP = 1$, $FN = 16$), and 55 ($TP = 55$, $FP = 0$, $FN = 6$) tassels in Figure 10a–c, respectively. This illustrates how the detection accuracy of highly overlapped objects is dependent on the size, and therefore the degree of overlap of the bounding boxes.

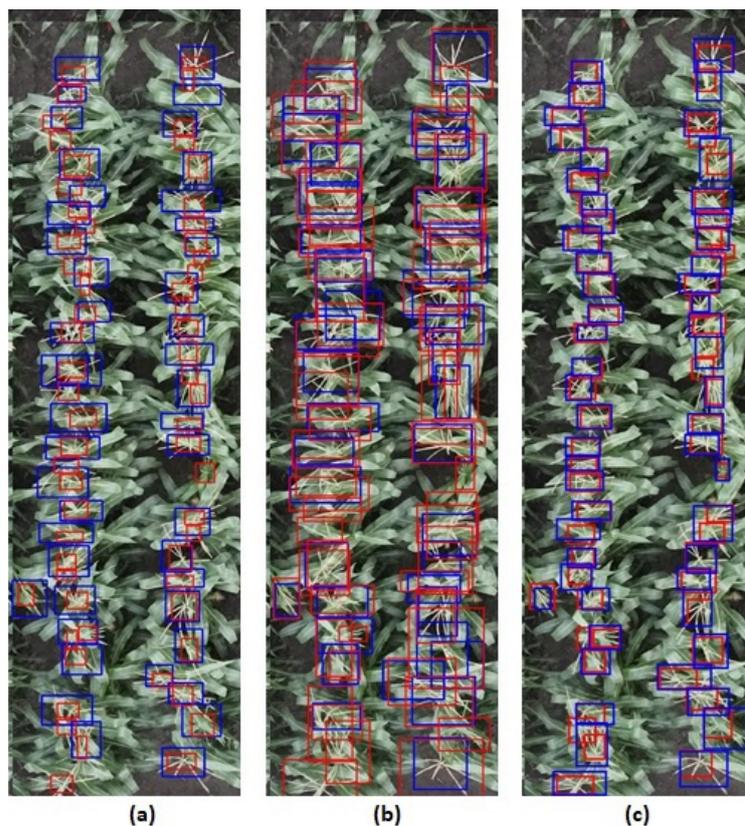


Figure 10. Bounding box annotation with different sizes (Red: Ground truth, Blue: Detected tassels); (a) Small; (b) Large; (c) Correct size.

3.3. Sensitivity to Tassel Density and Heterogeneity

As noted in Section 2, the dataset for tassel detection and counting was collected during the 2020 growing season, consisting of 88 panels with two replicas of inbred and hybrid varieties. In Figure 11, the image of the field on the west shows clear differences between the inbred panel planted in the north and the hybrids planted in the south side of the field. The canopy of the inbreds was less dense, and the tassels’ shapes, colors, structures, and stages of maturity differed from the hybrids as seen in Figure 12. The dataset provided an opportunity to evaluate the performance of the algorithms over a field with diverse tassel characteristics.

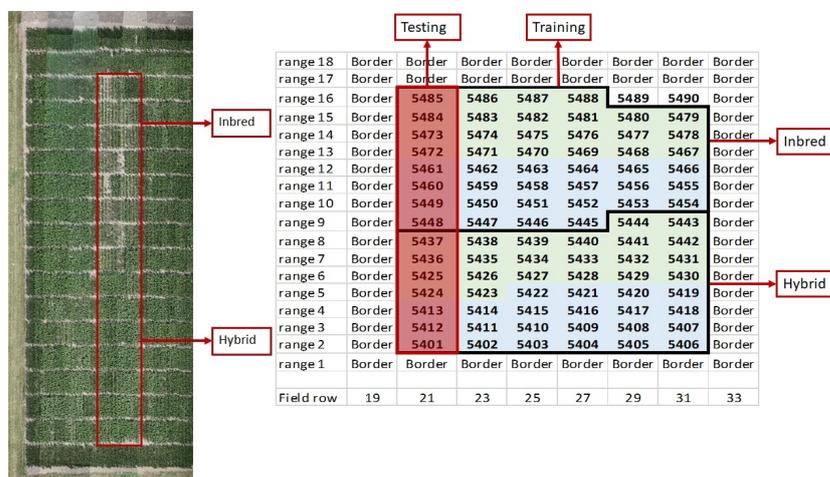


Figure 11. RGB image of the field and experiment layout. The training and testing regions were selected to include inbred and hybrid varieties.

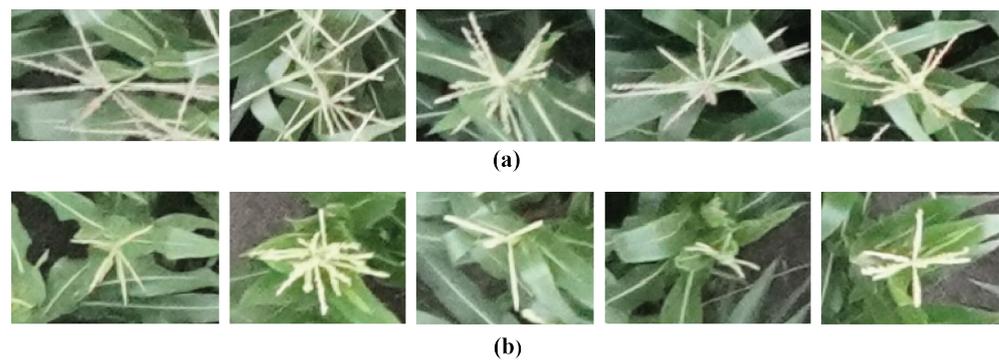


Figure 12. Tassels at different stages of maturity for (a) hybrid varieties and (b) inbred varieties.

3.4. Training and Testing Information

As described in Section 2, the original orthophoto was divided into 15 subsets (S1–S15) with a size of $\sim(3600 \times 2100)$ pixels. Two-row segments of plots on the west side of the field were used for testing, while the east side of the image was used for training. Plots S1–S7 correspond to hybrid entries and S8–S15 to inbreds. Therefore, data from both the hybrid and inbred varieties were included in both training and testing.

The number of training and testing images is shown in Table 2. The sizes of the training and test datasets differ for the three algorithms. This is primarily due to the requirements of the algorithms. If the size of the training images is close to testing images, the accuracy of tassel detection is artificially increased. TSD and DetectoRS can use different size datasets for training, unlike CenterNet architecture that requires (512×512) inputs. The image size $\sim(600 \times 2100)$ was used for training. The combined size of the training region is $\sim(3000 \times 2100)$. If this region is divided into (600×2100) , only five training images could be extracted from each subset (total 75), which is not adequate for training. A random crop with overlap was applied to the training region to extract 105 images. After that, the training images were randomly divided 90% and 10% into training and validation.

Table 2. Training and testing information.

Technique	Image Size	No. Training	No. Validation	No. Test	Training Time
TasselNetv2+	2100×600	97	8	15	1 h & 23 min
CenterNet	512×512	350	30	15	7 h & 34 min
TSD	2100×600	97	8	15	8 h & 41 min
DetectoRS	2100×600	97	8	15	7 h & 57 min

As mentioned, the CenterNet architecture requires (512×512) inputs. Thus, if the image coverage was selected similar to TSD and DetectoRS (2000×600), it would need to be resized to (512×512) . Because of the complexity in the tassel shapes, CenterNet could not train well. To mitigate the impact, a few-shot learning strategy similar to the idea which is used for plant counting in [5] was used to reduce the number of required training images. Finally, for CenterNet, the total number of training and validation images considered was 350 and 30, respectively. For a fair comparison, the size (512×512) was also considered for the training images of TSD and DetectoRS. However, this was so small that some of the annotated bounding boxes were not completely located in the training images not included during training (especially the tassels which are close to the image boundary). Because CenterNet uses point annotation, the number of missing tassels that were close to the boundary during the training was much lower. In the end, ~ 91 million pixels were selected for CenterNet, and ~ 122 million pixels were chosen for training TSD and DetectoRS, the training time being approximately equivalent for the three methods (see Table 2).

3.5. Comparison for Different Annotation Techniques

The detected tassels and ground test reference of five subsets are depicted in Figures 13–17 as examples. Figures 13 and 14 are examples of individual results for hybrid entries with fully emerged tassels (after the flowering date). Figures 15 and 16 depict inbred varieties with fully developed tassels (before the flowering date). For these subsets, CenterNet had the overall best performance based on the score of 99.15%, 94.73%, 95.15%, and 93.97%, respectively.

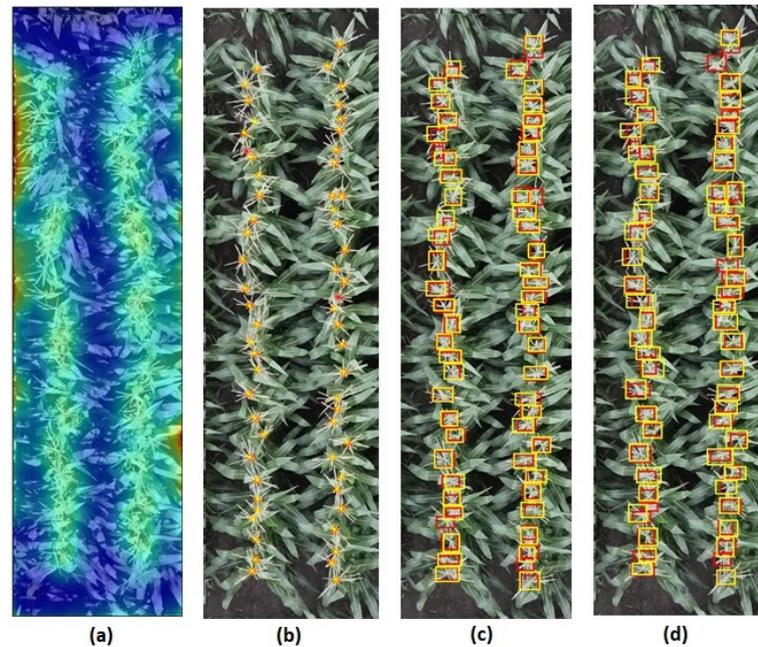


Figure 13. Results of Test Subset 4 (Ng: Red, TP: Yellow, FP: Cyan). (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetectoRS.

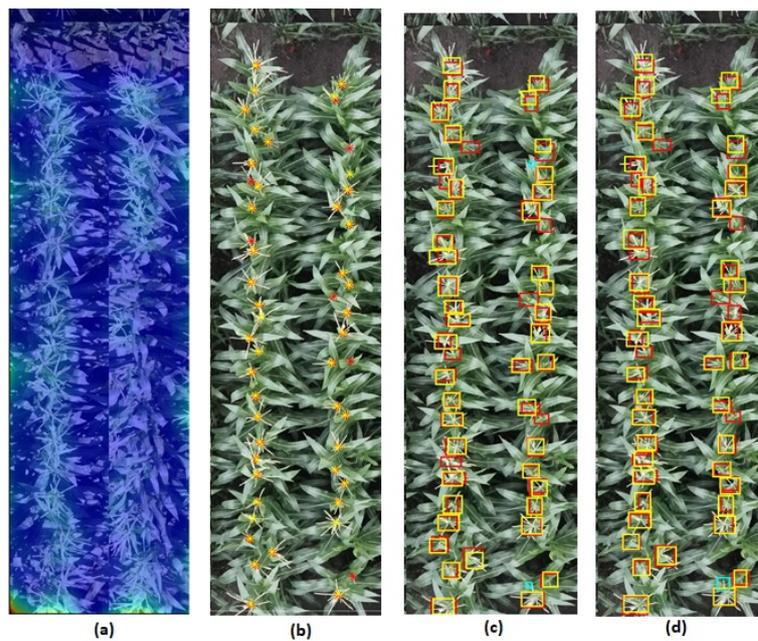


Figure 14. Results of Test Subset 5 (Ng: Red, TP: Yellow, FP: Cyan). (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetectoRS.

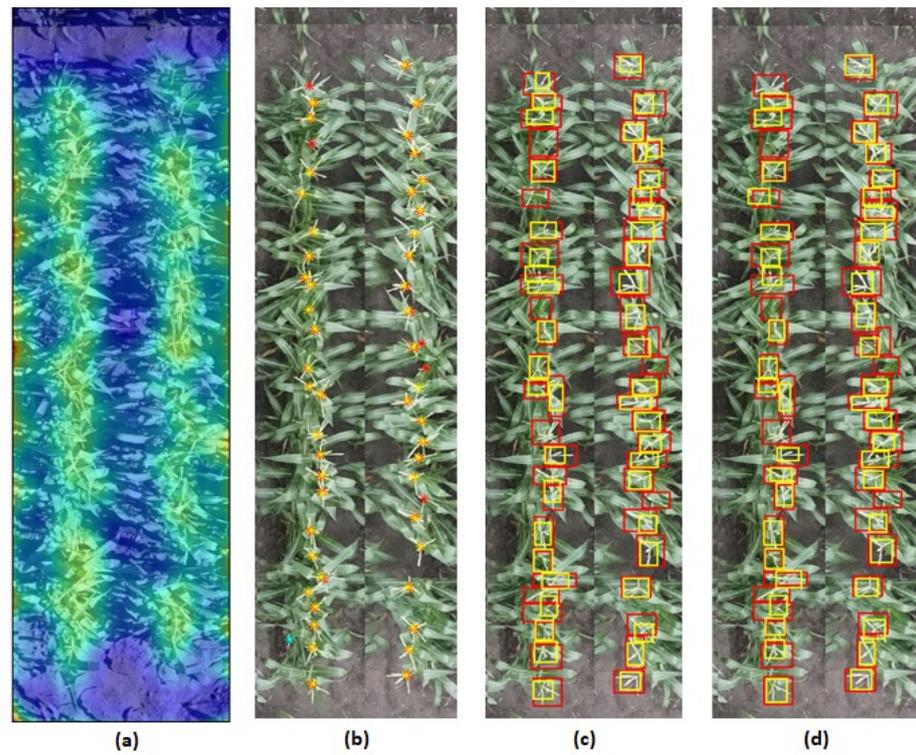


Figure 15. Results of Test Subset 12 (Ng: Red, TP: Yellow, FP: Cyan). (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetectoRS.

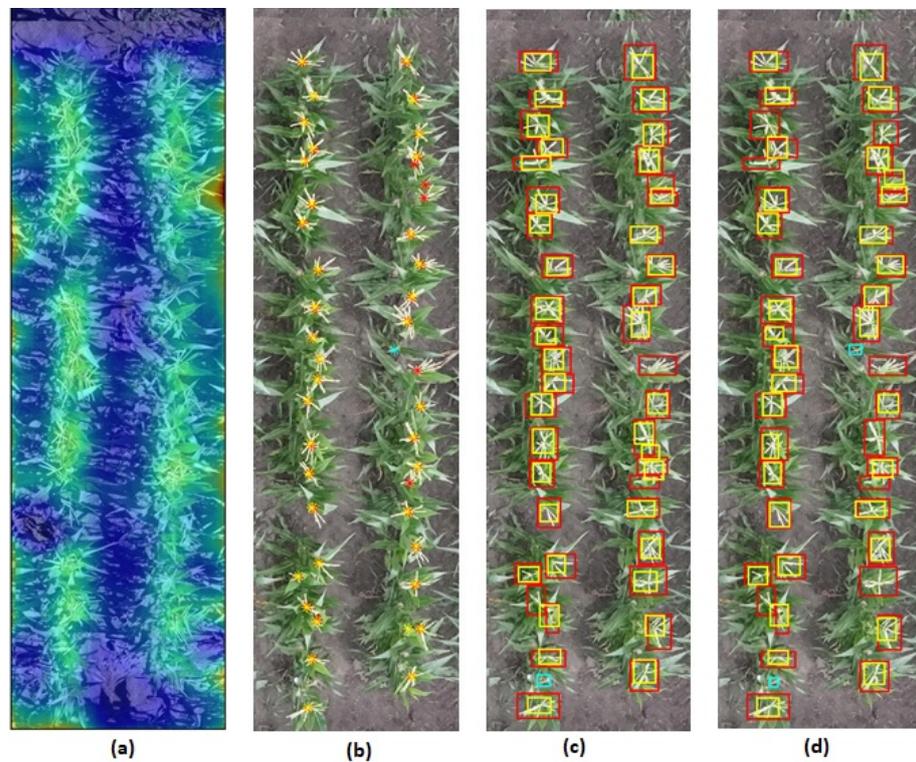


Figure 16. Results of Test Subset 15 (Ng: Red, TP: Yellow, FP: Cyan). (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetectoRS.

Subset 11 in Figure 17 is included to demonstrate the detection in plots where tassels had not yet emerged (before the flowering date). The ground truth indicated two tassels.

DetectorRS and TSD could only find one tassel, and CenterNet and TSD had false positives. The worst results were from TasselNetv2+, which incorrectly counted 29 tassels.

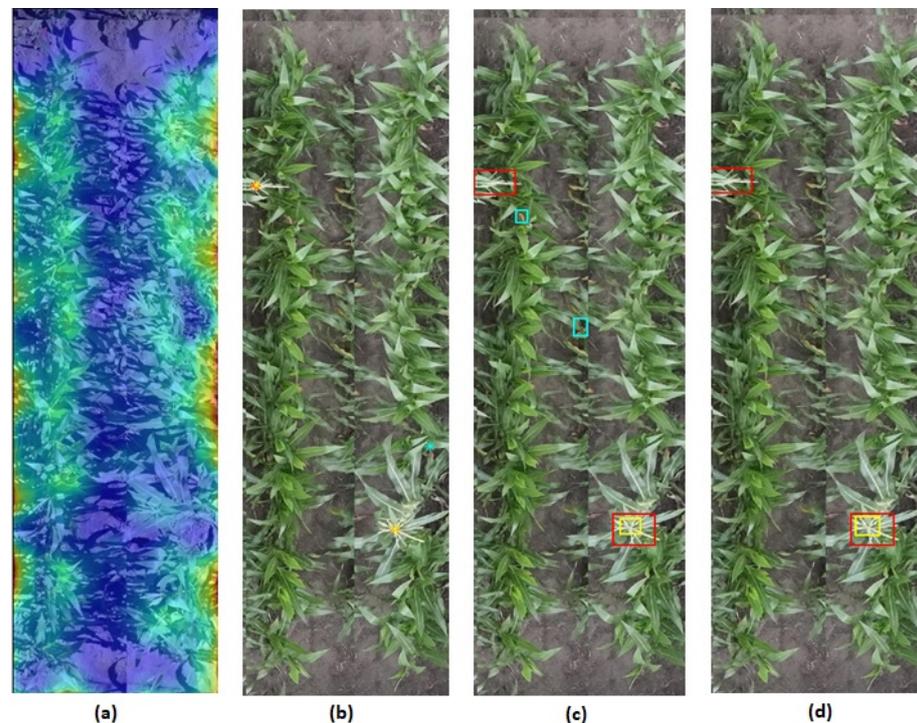


Figure 17. Results of Test Subset 11 (Ng: Red, TP: Yellow, FP: Cyan). (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetectorRS.

The performances of TasselNetv2+, CenterNet, DetectorRS, and TSD are shown in Tables 3–5. The results of these tables indicate that the tassels are detected more accurately by the detection approaches (CenterNet, TSD, and DetectorRS) than the regression approach (TasselNetv2+). As previously mentioned, inbred varieties had fewer tassels compared to the hybrids at the time of the data acquisition, and the performance of each detector is affected by the density of the tassels. Table 3 indicates that among the three methods, TSD had the lowest performance with a score of 89.90 and a standard deviation of 14.59.

Table 3. Testing results for detection-based metrics.

Technique	Metric	TP	FP	FN	Nt	Ng	Pr	Re	SC
CenterNet	Mean	38.67	0.40	3.07	39.13	41.93	96.86	90.97	93.24
	Std Dev	21.36	0.51	2.74	21.25	22.17	8.75	9.24	6.46
TSD	Mean	37.60	1.00	4.20	38.60	41.60	93.47	87.02	89.90
	Std Dev	20.38	1.41	2.88	20.76	22.86	17.49	11.42	14.59
DetectorRS	Mean	35.67	0.53	5.60	36.20	41.60	98.76	83.86	90.30
	Std Dev	19.33	0.74	3.14	19.64	22.03	1.79	10.33	7.15

The counting results in Table 4 show that TasselNetv2+ had the highest MAE and MSE values. Its performance is also significantly worse when the number of tassels in the image is low.

Table 4. Testing results for counting-based metrics.

Method	MAE	RMSE
TasselNetv2+	8.628	77.88
CenterNet	3.333	12.91
TSD	3.270	13.43
DetectoRS	5.400	20.94

Table 5 shows the scores for each testing plot using CenterNet, DetectoRS, and TSD. In most subsets, CenterNet obtained the best score value.

Table 5. Testing results for detection-based metrics for individual plots.

Testing Plot	Flowering Date	Ng	CenterNet	TSD	DetectoRS
S1	13-July	61	92.72	89.43	90.56
S2	13-July	62	99.18	95.93	95.86
S3	17-July	49	96.90	92.30	89.88
S4	18-July	61	99.15	96.60	94.01
S5	20-July	51	94.73	89.79	90.52
S6	17-July	58	100.00	91.84	92.72
S7	15-July	56	97.43	97.34	93.33
S8	22-July	22	89.99	92.68	93.76
S9	9-July	52	89.35	92.92	91.08
S10	22-July	46	96.83	97.67	93.82
S11	16-July	2	80.00	39.99	66.66
S12	26-July	53	95.14	91.99	91.83
S13	20-July	0	-	-	-
S14	25-July	9	79.99	94.11	95.37
S15	22-July	42	93.97	95.11	87.17

The details of tassel counting (TP , FP , and FN) for each of the fifteen subsets are shown in Figure 18. The number of manually counted tassels (Ng) was always somewhat larger than the number detected by the algorithms, although the difference between TP and Ng was not statistically different for any algorithm. CenterNet had the maximum average value of TP . TSD had the largest average value of FP , and DetectoRS had the largest average FN value.

As shown in Figure 19, the mean precision value of DetectoRS is higher than CenterNet and TSD. Investigating further, Table 5 shows that for S13, ($Ng = 0$). Therefore, Figure 19 does not have any information for this subset. The standard deviation of the precision of DetectoRS is also small. We can infer the DetectoRS could detect the actual tassels well. However, the recall value is not as high as CenterNet. The accuracy of TSD is lower than CenterNet, and it detects more false positives (FP) as incorrect tassels. Furthermore, creating the bounding box around the tassels for training DetectoRS and TSD is time-consuming, and these algorithms are sensitive to the size of bounding boxes.

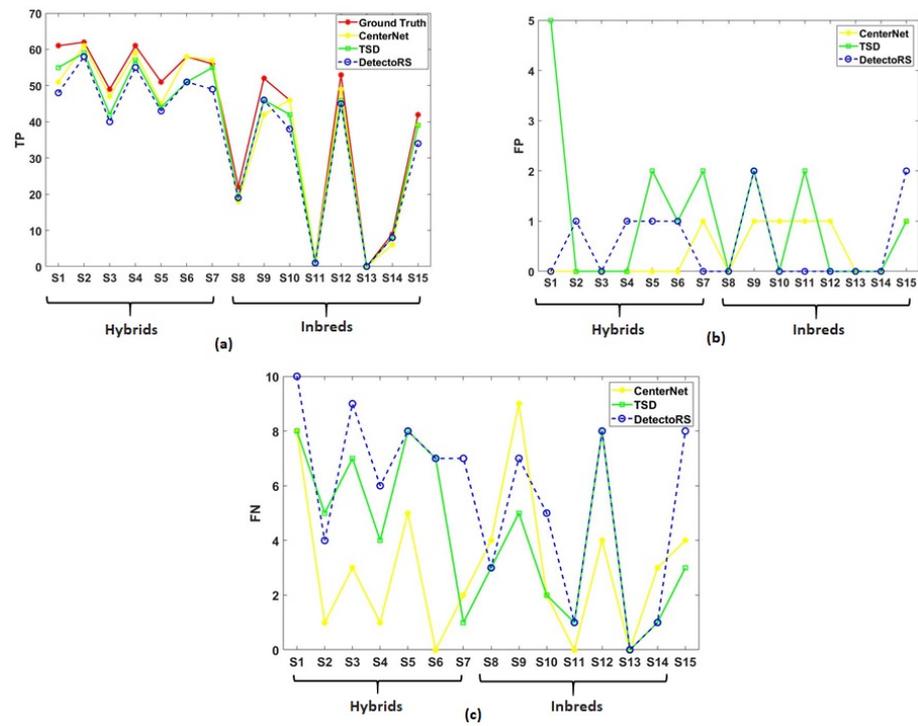


Figure 18. Results of Test (Ng: Red, CenterNet: Yellow, TSD: Green, DetecoRS: Blue). (a) TP. (b) FP. (c) FN.

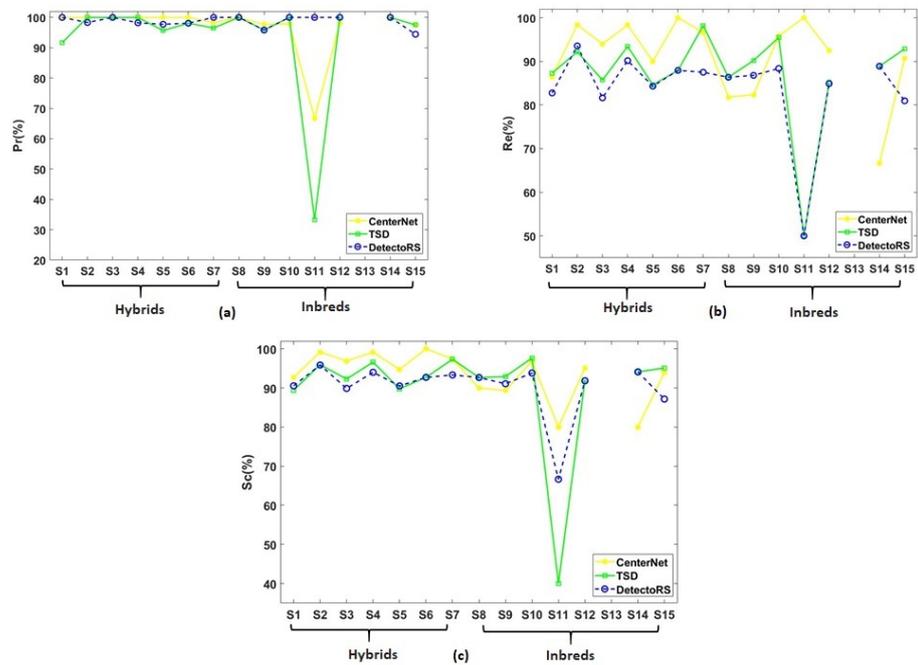


Figure 19. Results of Test (CenterNet: Yellow, TSD: Green, DetecoRS: Blue). (a) Precision (%). (b) Recall (%). (c) Score (%).

The linear regressions between the manual counting and the predicted number of tassels using the three detection-based methods (CenterNet, DetecoRS, and TSD) and one regression-based method (Tasselnetv2+) are shown in Figure 20.

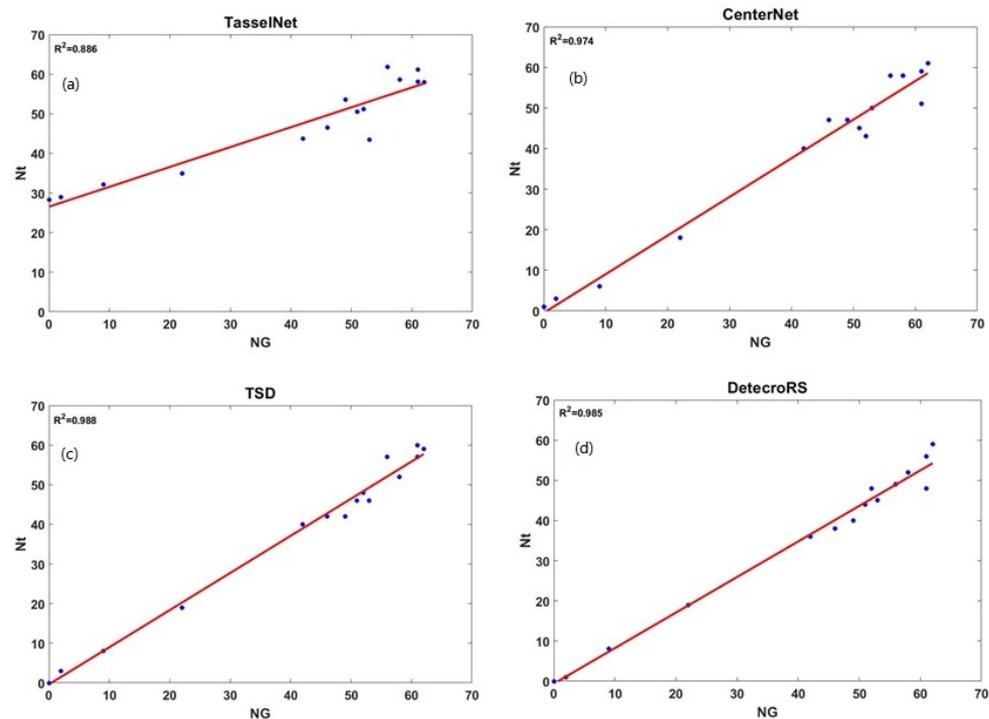


Figure 20. Linear regression plots for (a) TasselNetv2+. (b) CenterNet. (c) TSD. (d) DetecroRS.

The TSD and DetecroRS techniques provide slightly higher fidelity counts than CenterNet and Tasselnetv2+ (see Figure 20). However, CenterNet has the highest mean value for TP and the lowest FP and FN values. Because of the date of the imagery, most plots were in the mid-to-late stages of flowering and had a large number of tassels. TasselNetv2+ could only provide good results when the number of tassels in the subset was very high. The impact of overcounting by TasselNetv2+ compared to ground reference data in plots with a small number of tassels is clearly visible.

4. Conclusions

A key goal of this study was to investigate the value of detection-based approaches compared to regression counting methods in the complex scenario of in-field tassel counting. In this article, three state-of-the-art object detection algorithms, CenterNet, DetecroRS, and TSD, were modified for tassel detection and compared to counts obtained by TasselNetv2+, as well as image-based ground reference counts. All three algorithms had good overall performance in terms of the number of true positives compared to the ground reference. CenterNet achieved the highest recall value and score, and DetecroRS obtained the highest value of precision. TSD has the lowest score value. The performance of TasselNetv2+, which only provides information about counting, is highly dependent on the number of tassels.

Specific annotation is used for each of the detection-based algorithms. Two types of label annotations, “point” and “bounding boxes”, were investigated. DetecroRS and TSD require bounding box annotations, and results are sensitive to the size of the bounding boxes. CenterNet works based on point annotation, which is simpler, faster to collect, and more accurate than bounding boxes

As future work, the next steps will explicitly consider multiple dates during the flowering process, where there is greater diversity in the number and characteristics of the developing tassels. The strategy will investigate multitemporal analysis and attention-based networks to predict flowering (anthesis) dates. The impact of temperature will also be investigated via the incorporation of growing degree days (GDD).

Author Contributions: Data collection and post-processing, K.Q.; Data annotation, A.K. and K.Q.; Experimental process design, A.K.; Writing—original draft, A.K. and K.Q., Writing—review and editing, M.C., A.K., and K.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000593. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code from this paper are available on request from the corresponding author (Karami). The data presented in this study are available on request from Author 3 (Crawford).

Acknowledgments: The authors thank the Purdue TERRA team, especially Taojun Wang, Yi-Chun Lin, Meghdad Hasheminasab, and Tian Zhou, for their work in data collection and processing, and Claudia Aviles, Changye Yang, Enyu Cai, and Jethro Zhou for their contributions to data annotation. They are also appreciative of the valuable input provided by Ayman Habib, Edward Delp, and Mitchell Tuinstra throughout this work.

Conflicts of Interest: The authors declare no conflict of interest

References

- Ji, M.; Yang, Y.; Zheng, Y.; Zhu, Q.; Huang, M.; Guo, Y. In-field automatic detection of maize tassels using computer vision. *Inf. Process. Agric.* **2021**, *8*, 87–95.
- Su, Y.; Wu, F.; Ao, Z.; Jin, S.; Qin, F.; Liu, B.; Pang, S.; Liu, L.; Guo, Q. Evaluating maize phenotype dynamics under drought stress using terrestrial LiDAR. *Plant Methods* **2019**, *15*, 1–16. [[CrossRef](#)] [[PubMed](#)]
- Guo, W.; Fukatsu, T.; Ninomiya, S. Automated characterization of flowering dynamics in rice using field-acquired time-series RGB images. *Plant Methods* **2015**, *11*, 7. [[CrossRef](#)]
- Mirnezami, S.V.; Srinivasan, S.; Zhou, Y.; Schnable, P.S.; Ganapathysubramanian, B. Detection of the progression of anthesis in field-grown maize tassels: A case study. *Plant Phenomics* **2021**, *2021*, 4238701. [[CrossRef](#)]
- Karami, A.; Crawford, M.; Delp, E.J. Automatic plant counting and location based on a few-shot learning technique. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5872–5886. [[CrossRef](#)]
- Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process. Agric.* **2020**, *7*, 1–19. [[CrossRef](#)]
- Gage, J.L.; Miller, N.D.; Spalding, E.P.; Kaeppler, S.M.; de Leon, N. TIPS: A system for automated image-based phenotyping of maize tassels. *Plant Methods* **2017**, *13*, 1–12. [[CrossRef](#)] [[PubMed](#)]
- Ye, M.; Cao, Z.; Yu, Z. An image-based approach for automatic detecting tasseling stage of maize using spatio-temporal saliency. In *Remote Sensing Image Processing, Geographic Information Systems, International Society for Optics and Photonics*; SPIE: Washington, DC, USA, 2013; Volume 8921, p. 89210Z.
- Kurtulmuş, F.; Kavdir, I. Detecting corn tassels using computer vision and support vector machines. *Expert Syst. Appl.* **2014**, *41*, 7390–7397. [[CrossRef](#)]
- Oscro, L.P.; Junior, J.M.; Ramos, A.P.M.; Jorge, L.A.C.; Fatholahi, S.N.; Silva, J.A.; Matsubara, E.T.; Gonçalves, W.N.; Pistori, H.; Li, J. A review on deep learning in UAV remote sensing. *arXiv* **2021**, arXiv:2101.10861.
- Karami, A.; Crawford, M.; Delp, E.J. A weakly supervised deep learning approach for plant center detection and counting. In *Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020*; pp. 1584–1587.
- Valente, J.; Sari, B.; Kooistra, L.; Kramer, H.; Múcher, S. Automated crop plant counting from very high-resolution aerial imagery. *Precis. Agric.* **2020**, *21*, 1366–1384. [[CrossRef](#)]
- Hasan, M.M.; Chopin, J.P.; Laga, H.; Miklavcic, S.J. Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods* **2018**, *14*, 1–13. [[CrossRef](#)] [[PubMed](#)]
- Ghosal, S.; Zheng, B.; Chapman, S.C.; Potgieter, A.B.; Jordan, D.R.; Wang, X.; Singh, A.K.; Singh, A.; Hirafuji, M.; Ninomiya, S.; et al. A weakly supervised deep learning framework for sorghum head detection and counting. *Plant Phenomics* **2019**, *2019*, 1525874. [[CrossRef](#)] [[PubMed](#)]
- Yang, K.; Zhong, W.; Li, F. Leaf segmentation and classification with a complicated background using deep learning. *Agronomy* **2020**, *10*, 1721. [[CrossRef](#)]
- Zou, H.; Lu, H.; Li, Y.; Liu, L.; Cao, Z. Maize tassels detection: A benchmark of the state of the art. *Plant Methods* **2020**, *16*, 108. [[CrossRef](#)] [[PubMed](#)]
- Lu, H.; Cao, Z.; Xiao, Y.; Zhuang, B.; Shen, C. TasselNet: Counting maize tassels in the wild via local counts regression network. *Plant Methods* **2017**, *13*, 79. [[CrossRef](#)]

18. Xiong, H.; Cao, Z.; Lu, H.; Madec, S.; Liu, L.; Shen, C. TasselNetv2: In-field counting of wheat spikes with context-augmented local regression networks. *Plant Methods* **2019**, *15*, 150. [[CrossRef](#)]
19. Lu, H.; Cao, Z. TasselNetV2+: A fast implementation for high-throughput plant counting from high-resolution RGB imagery. *Front. Plant Sci.* **2020**, *11*, 1929. [[CrossRef](#)]
20. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.
21. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In Proceedings of the 2020 Conference on Neural Information Processing Systems, NeurIPS, Vancouver, BC, Canada, 21–22 December 2020.
22. Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss V2: Learning reliable localization quality estimation for dense object detection. *arXiv* **2020**, arXiv:2011.12885.
23. Shinya, Y. USB: Universal-scale object detection benchmark. *arXiv* **2021**, arXiv:2103.14027.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [[CrossRef](#)]
25. Song, G.; Liu, Y.; Wang, X. Revisiting the sibling had in object detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11560–11569. [[CrossRef](#)]
26. Sun, Z.; Cao, S.; Yang, Y.; Kitani, K. Rethinking transformer-based set prediction for object detection. *arXiv* **2020**, arXiv:2011.10881.
27. Qiao, S.; Chen, L.C.; Yuille, A. DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv* **2020**, arXiv:2006.02334.
28. Liu, Y.; Cen, C.; Che, Y.; Ke, R.; Ma, Y.; Ma, Y. Detection of maize tassels from UAV RGB imagery with faster R-CNN. *Remote Sens.* **2020**, *12*, 338. [[CrossRef](#)]
29. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. Faceboxes: A CPU real-time face detector with high accuracy. In Proceedings of the International Joint Conference on Biometrics, Denver, CO, USA, 1–4 October 2017; pp. 1–9.
31. Kumar, A.; Taparia, M.; Rajalakshmi, P.; Desai, U.; Naik, B.; Guo, W. UAV based remote sensing for tassel detection and growth stage estimation of maize crop using F-RCNN. *Comput. Vis. Probl. Plant Phenotyping* **2019**, *3*, 4321–4323.
32. Lu, X.; Li, B.; Yue, Y.; Li, Q.; Yan, J. Grid R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7363–7372.
33. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. Foveabox: Beyond anchor-based object detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [[CrossRef](#)]
34. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 734–750.
35. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 850–859.
36. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
37. Lin, Y.C.; Zhou, T.; Wang, T.; Crawford, M.; Habib, A. New orthophoto generation strategies from UAV and ground remote sensing platforms for high-throughput phenotyping. *Remote Sens.* **2021**, *13*, 860. [[CrossRef](#)]
38. The Genomes to Fields Initiative (G2F). Standard Operating Procedures (SOP). 2020. Available online: <https://www.genomes2fields.org/resources/> (accessed on 22 April 2021).
39. Wada, K. Labelme: Image Polygonal Annotation with Python. 2018. Available online: <https://github.com/wkentaro/labelme> (accessed on 9 May 2016)
40. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *arXiv* **2019**, arXiv:1906.09756.
41. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 764–773.
42. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected CRFS. *arXiv* **2014**, arXiv:1412.7062.
43. Papandreou, G.; Kokkinos, I.; Savalle, P.A. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 390–399.
44. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.